Visual Servoing in Robotic Manufacturing Systems for Accurate Positioning

Zheng Li

A Thesis
In The Department of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

January, 2007

Your file Votre référence
ISBN: 978-0-494-28941-9
Our file Notre référence
ISBN: 978-0-494-28941-9

Abstract

Visual Servoing in Robotic Manufacturing Systems for Accurate Positioning

Zheng Li

Automated robotic manufacturing systems require accurate robot positioning. Visual servoing is an increasing popular method to enhance such positioning accuracy. Based on the error signal definition, visual servoing is classified into three approaches, Position Based Visual Servoing (PBVS), Image Based Visual Servoing (IBVS) and Hybrid Visual Servoing (HVS)

In this research, firstly, a novel Neural Network (NN) based hand-eye calibration is introduced in PBVS. A MultiLayer Perceptron NN is used to approximate the nonlinear coordinate transform from image coordinates to real world coordinates in visual servoing. The main advantages of NN based hand-eye calibration are that it can solve the hand-eye calibration problem without estimating the hand-eye transformation and can improve the object tracking accuracy as well. The experimental results in an industrial manufacturing robot show that the proposed calibration method outperforms the current solving transformation matrix method and free hand-eye calibration method for 2D object tracking.

Secondly, a new approach to switching control of IBVS with laser pointer is proposed. The simple off-the-shelf laser pointer is applied to realize the depth estimation. The proposed system is robust to the camera calibration and hand-eye calibration error, and is object model free as well. Comparing with traditional IBVS, it avoids image singularities

and image local minima, and is successful for only partial image features in the field of view. Moreover, the trajectory of the robot end effector is shortened. The experimental results are given to verify the effectiveness of the proposed method in a robotic manufacturing system for assembly.

Acknowledgments

I would like to express my sincere appreciation to my supervisor, Dr. Wen-Fang Xie. It is her knowledge, patience and help that guided me throughout my research. I am grateful to my supervisor Dr. Xiao-Wei Tu for his invaluable support and guidance for the implementation of the project. I would also like to express my thanks to my previous supervisor Dr. Nabil Aouf, who leads me to the visual servoing.

I am extremely grateful to Aerospace Manufacturing Technology Centre (AMTC), National Research Council Canada (NRC), which has provided facilities for my research and such a great opportunity for me to get into the amazing visual servoing field. In addition, I would like to thank the robotic group of AMTC for their continuous support and help.

I would also like to express my appreciations to all colleagues in the lab of Concordia University, especially Zhao Zhongyu who provided several useful suggestions on my research.

I would like to take this opportunity to thank everyone who has helped me during the completion of this thesis.

My families have been enormously supportive for rather a long period of time. I dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURES

DOF	Degree of freedom
B	Robot reference frame
E	Robot end effector frame
W	World reference frame
C	Camera reference frame
O	Object reference frame
I	Image reference frame
$\mathbf{A}$	Matrix
${}^{F_1}\mathbf{H}_{F_2}$	Homogeneous transformation matrix of the reference frame F_2 with respect to the reference frame F_1.
${}^{F_1}\mathbf{R}_{F_2}$	Rotation matrix of the reference frame F_2 with respect to the reference frame F_1.
${}^{F_1}v_{F_2}$	Translational velocity vector of the reference frame F_2 with respect to the reference frame F_1.
${}^{F_1}\omega_{F_2}$	Angular velocity vector of the reference frame F_2 with respect to the reference frame F_1.
${}^{F_1}p_{F_2}$	A point vector of the reference frame F_2 with respect to the reference frame F_1.

${}^{F_1}\dot{p}_{F_2}$	A point velocity of the reference frame F_2 with respect to the reference frame F_1.
Sk	Skew symmetric matrix
λ	Focal length
(u_0, v_0)	Image center pixel coordinates (principal point)
(u, v)	Pixel index of a point
(x_i, y_i)	Image coordinates of a point in camera reference frame
k_1	The lens distortion coefficient
s_x	Effective size of the pixel in the horizontal direction
s_y	Effective size of the pixel in the vertical direction
$\dot{r}$	Velocity screw of the camera
K	Proportional gain
f	Image feature
$\dot{f}$	Image feature velocity
Z	Depth of feature point
J	Jacobian matrix
$\mathbf{J}_{img}$	Image Jacobian matrix
T_x	A pure translation along X axis
T_y	A pure translation along Y axis
T_z	A pure translation along Z axis
ϕ	A pure rotation around the Z axis (roll)

θ	A pure rotation around the Y axis (pitch)
ψ	A pure rotation around the X axis (yaw)
$\mathbf{M}_{int}$	A matrix related to camera intrinsic parameters
d	The horizontal distance between the laser beam and the optical axis
α	The angle between the laser beam and the horizon.

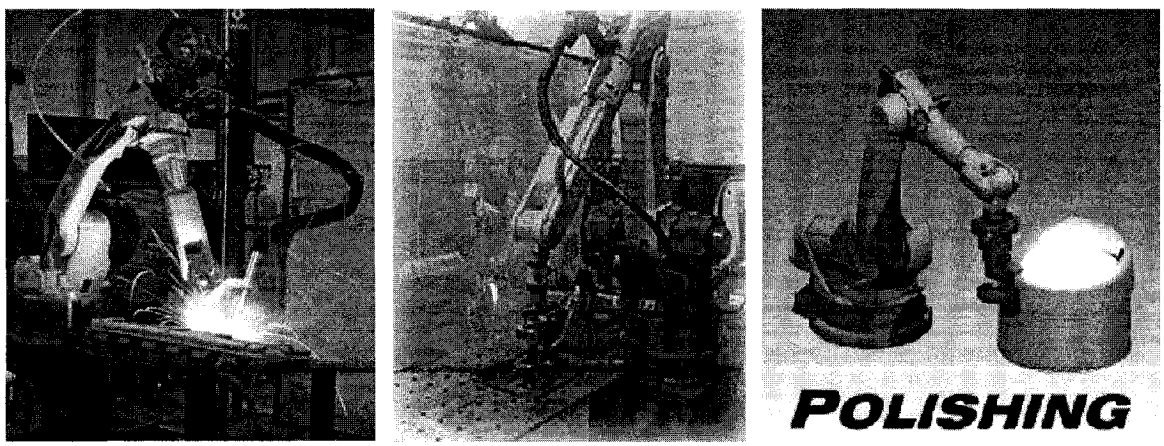
CHAPTER 1 INTRODUCTION

1.1 Introduction

The task in visual servoing is to use machine vision system to control the pose of the robot's end effector relative to a target object, thereby to compensate for the robotic positioning errors. This chapter introduces the basic concepts and components of visual servoing system, together with an overview of the background. The motivation, objective and contribution of this study are also summarized.

1.2 Background and Previous Work

Automated robotic manufacturing systems such as robotic assembly, drilling, welding, painting, shot peening, and surface finishing (deburring, polishing, grinding), all require accurate robot positioning [1]. Some pictures of robotic manufacturing system are shown in Figure 1-1.



(a) Welding[2] (b) Drilling[3] (c) Polishing

Figure 1-1 Robotic manufacturing systems

For a conventional robot manipulator, the accurate position is only derived from the feedback control system of the joints. The repeatability may reach within ± 0.1 mm. However, several uncertainties may affect the precision, such as poor robot calibration and the changing of load. Furthermore, conventional robot does not have the ability of self teaching.

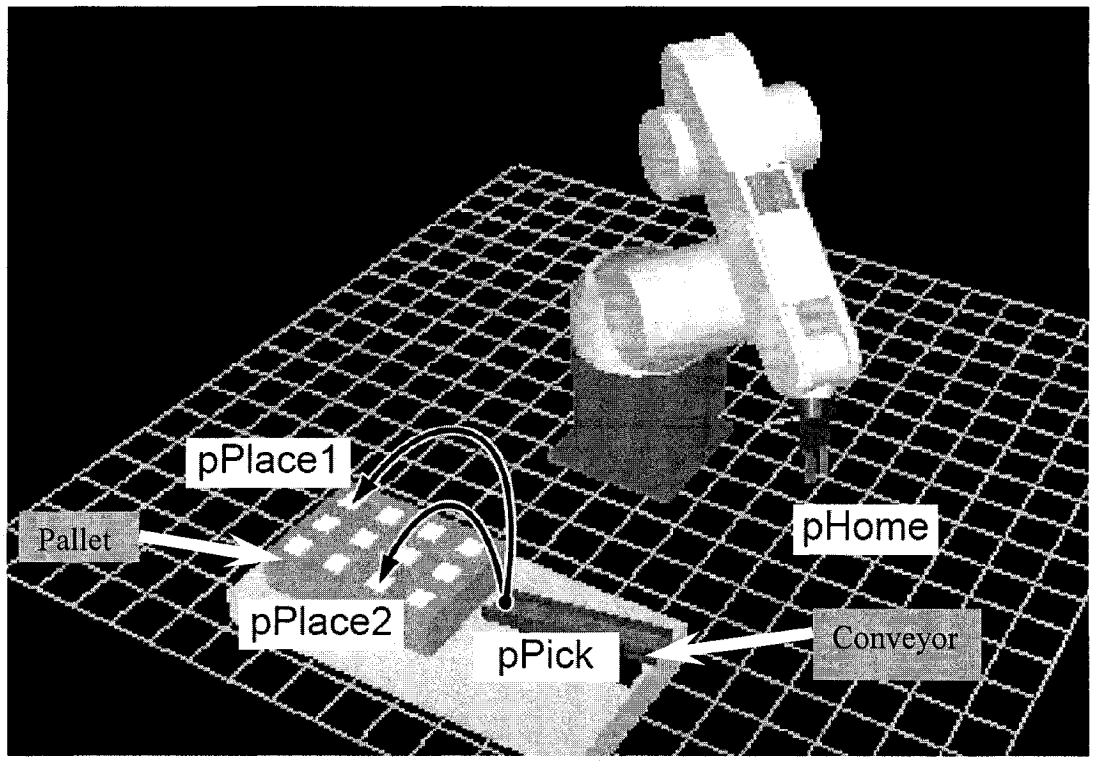


Figure 1-2 Robot palletizing operations [7]

Take the operation shown in Figure 1-2 as an example. The equipment setup with the robot manipulator is included as part of the production line. Before this robot performs palletizing operations, it is necessary to determine the values to be assigned to the respective position variables "pHome", "pPick", "pPlace1" and "pPlace2" for the program [7]. However, in case where either the position of the pallet or conveyor is changed to a new position, the robot will not achieve the task any more. Another example is related to robotic drilling. Suppose that there are hundreds of holes to be drilled on a

board. By using a conventional robot system, we have to teach robot the exact position of every hole to be drilled, which is very tedious and time consuming.

In these situations, a new type of sensor needs to be added to the conventional robot system. One of the effective sensors is visual sensor, such as camera. With the fast development of hardware and computing power, more and more industrial customer choose visual feedback to increase the robot positioning accuracy and realize some non-contact measurement of the environment as well. A robot manipulator together with a vision system can be used to imitate the motion of human beings. The accuracy of the resulting operation depends both on the accuracy of the visual sensor and the robot manipulator [4][5]. Take the robotic drilling with visual servoing as an example, a large number of marks are preinstalled or painted on the board at the manufacture phase. The vision system will detect the marks and guide the robot manipulator to achieve the drilling task. Although the vision system increases the cost, it improves the positioning efficiency, increases the accuracy and saves the manpower.

The term of visual servoing is first introduced in [9], which denotes that the vision based control system can provide closed loop position control for robot end effector. It is the fusion of many related areas including image processing, kinematics, dynamics, real-time computing, machine vision, and control theory [4][5]. The related background knowledge and previous related works of visual servoing are introduced in this section, including the classification of visual servoing and a review of hand-eye calibration.

1.2.1 Classification of Visual Servoing

There are several types of architecture of visual servoing system, depending on the classification method.

According to the position of the camera, it can be classified into two architectures: eye in hand while the camera is rigidly mounted on the robot end effector to observe the object only, and eye to hand while it is fixed in the workspace to observe both the robot and the object [10][11]. Here, hand is used to denote robot end effector, and eye is the camera. Obviously, the transformation between hand and eye is constant in architecture one. On the contrary, the transformations from eye to world reference frame and stable object frame are constant in architecture two. In [4], authors present the first architecture as endpoint open loop (EOL) system, and endpoint closed loop (ECL) system. Although the ECL system will give the relative accurate results, the losing field of view problem cannot be solved.

The second major classification of visual servoing distinguishes direct visual servoing from look-and-move architectures [12][13]. As the name implies, direct visual servo controller directly computes the torque inputs of joints, hence it uses the vision feedback alone to stabilize the whole system. The robot dynamics needs to be considered. In contrast, the look-and-move structure uses the vision based controller to generate the inputs of the joint controller of robot. Therefore the stability of the whole visual servoing system is dependent on both the low level robot joint controller and the visual controller. There are many advantages of look-and-move approach. As we know, the vision system is hard to provide a high sample rate because of the limitation of camera. The low level joint controller with a high sample rate can easily satisfied the stability of the robot, thus a lower sample rate of the outer loop, vision feedback, can still fulfill the task. Moreover, most of commercial robots do not provide the customers access to the joint control, and already have the controller of joints that makes the visual servo system simple and

shortens the developing period [4]. In the research, we adopt the look-and-move structure exclusively, and in this thesis, all mentioned visual servoing denotes look-and-move structure.

The last classification of visual servoing provided here is dependent on the error signal definition. The error for Position Based Visual Servoing (PBVS) is defined in 3D Cartesian space, and the error for Image Based Visual Servoing (IBVS) is defined in image plane. Hybrid Visual Servoing (HVS) is to use a hybrid of Cartesian and image space sensory feedback signals to control both the Cartesian and image trajectories simultaneously [4][6]. The advantages and disadvantages of PBVS, IBVS and HVS are given in following sections in detail. Some performance tests for visual servoing control systems are provided in [34][35][39].

- **PBVS**

During the servoing, PBVS calculates the relative position and orientation, which are known as the pose of the object with respect to camera. The error between the desired and current pose is defined in 3D Cartesian space. The robot end effector trajectory is controlled to move approximately along straight lines in the Cartesian space. Because PBVS separates the pose estimation from the controller design, it allows the integration of measurements from multiple sensors. However, it needs accurate camera calibration and hand-eye calibration, and perfect target geometric model to achieve the pose [14][15]. Furthermore, since there is no control of the image trajectories, the image boundary constraint may be violated, which may lead to potential failure of servoing.

➢ **IBVS**

Since IBVS was introduced in 1980, it attracts the attention of many researchers and has been tremendously developed in recent years. The method is based on the principle that when the image feature error is approaching to zero, the kinematic error will approach to zero too. In IBVS the error for the controller is defined directly with respect to image feature parameters. Comparing with PBVS, the advantages of IBVS are obvious. First it is object model free, and robust with respect to camera modeling and hand-eye calibration errors [84]. Second the image feature point trajectories are controlled to move approximately along straight lines in the image plane so that it is able to prevent the image features from leaving the Field of View (FOV). However, the drawbacks of IBVS still exist. Since the control law is merely in the image plane, the trajectory of the end effector is not a straight line any more, and even odd in some cases. In other words, in order to make the image feature error reduce to zero as soon as possible, unnecessary motions of end effector are performed. Moreover it is stable only in a region around the desired position, and there may have image singularities and image local minima leading to potential failure [16][17]. Image local minima are first introduced in [16], which may lead to IBVS failure. Koichi Hashimoto et al [62] introduced a potential method to solve the image local minima. The main idea of their algorithm is minimizing the potential function. If the potential has local minima, the algorithm generates an artificial potential and then controls the camera base on the artificial one.

To avoid the weakness of IBVS and achieve a more functional visual servoing system, researchers have provided some improved algorithms. In [19], to solve the depth estimation problem and calculate an exact image Jacobian matrix, stereo based visual

servoing is proposed. However this kind of system increases the computational cost. R.Mahony et al [20] introduced a method of choosing other type of image features instead of point for IBVS and focusing on the depth axis control. P.Y.Oh et al [21] presents partitioning DOF method for IBVS. The experimental results are given by tracking people. In [22], another partitioned approach to visual servo control is introduced, which decouples the z-axis rotational and translational components of the control from the remaining DOF. The method in [23] is based on a cylindrical coordinate system that can shift the position of the origin.

Lately some active visual servoing systems are introduced. In [24], Pages et al. derive the image Jacobian matrix related laser spot. However, the method needs the information of the planar object. In their algorithm, it is only a fit for the situation that the camera is located near the desired position. Another approach using laser pointer in visual servoing is presented in [25][26]. Krupa et al. develop a vision system with static camera, which retrieves and positions surgical instruments for operation. A laser pointer is used to project laser spot on the organ surface to control the depth. However, the servoing is only 3DOF and the camera is motionless, so the system cannot provide much flexibility.

- **HVS**

As the shortcomings of PBVS and IBVS mentioned above, some new approaches that integrate the two methods are developed. The main idea is to use a hybrid of Cartesian and image space sensory feedback signals to control both the Cartesian and image trajectories simultaneously.

One of the hybrid visual servoing is 2.5D visual servoing [27][28][29]. Inheriting the merits of position based and image based methods, it needs neither target model nor

accurate camera calibration and hand-eye calibration. The Cartesian and image trajectories are controlled simultaneously so that they are approximate to straight line in 3D Cartesian space and image plane respectively. It avoids the image singularity and local minima. One of the drawbacks of this method is that it is more sensitive to image noise than IBVS, since this scheme directly uses visual features as input of the control law, without any supplementary estimation step [28]. Also note that only one of the image features is chosen for the image Jacobian matrix, which will be susceptible to noise. Another hybrid method is presented in [32][33], which uses a high level decision maker to select from two visual servo controllers: homography based controller and affine approximation controller. Recently, a hybrid motion control and planning strategy for image constraints avoidance are presented in [36][37][38]. This motion control part includes a local switching control between the IBVS and PBVS for avoiding image singularity and image local minima. In addition, the planning strategy is composed of an artificial hybrid trajectory planner.

In summary, the classification of visual servoing in the literature survey is shown in Table 1-1.

Table 1-1 Classification of visual servoing

Methods for Classification	Class
Position of the camera	Eye in hand
	Eye to hand
The control input signal	Direct visual servoing
	Look-and-move
Error signal definition	PBVS
	IBVS
	HVS

1.2.2 Review of Hand-Eye Calibration

In order to control the displacements of a robot by a camera mounted on its hand, it is important to determine the relative positions and orientations of the camera and hand frames. This problem is known as the hand-eye calibration problem [40][42].

Hand-eye calibration is an offline procedure and is the fundamental step of accurate PBVS. In other words, if hand-eye calibration was inaccurate, so would be the calculation of object position. Generally the hand-eye calibration problem is solved using techniques of camera calibration and robot motion control. As the preliminary step of hand-eye calibration, camera calibration is to estimate the value of extrinsic and intrinsic parameters of camera, which will be discussed in detail in Chapter 2. When the camera observing the object is fixed, standard camera calibration models, dependent on the camera position, can be obtained using least squares method [41]. When the camera is moving with the robot end effector in eye in hand configuration, there are some techniques to solve the hand-eye transform, such as solving for transformation equations of the form AX=XB and free hand-eye calibration based on the Tsai's camera calibration technique. Both of these methods will be introduced as follows in detail.

1.2.2.1 Solving AX=XB

Most of existing solutions for hand-eye calibration lead to solving for homogeneous transformation equations of the form AX=XB [42][43][44]or AX=ZB [45], where X and Z represent hand/eye and robot/world transformations respectively. A is a relative transform between two different end-effecter frames, and B is a relative transform between the two camera frames which is obtained from camera calibration.

In [44], a formalism of hand-eye calibration with a lens distortion model and an algorithm for optimizing intrinsic camera parameters using extended Kalman filter is introduced. A calibration pattern board is located at three depths, and at each depth, images are taken at five viewpoints. The intrinsic camera calibration parameters are optimized over the entire viewpoint extrinsic camera parameter set. The transformation matrices for hand-eye calibration, including the transformation from the camera to the end effector and the transformation from the robot base to the world reference frame, are estimated. By solving the equation of AX=XB and using the Broyden-fletcher-Goldfarb-Shanno optimization algorithm [46], rotational and translational components **R,T** of those transformations are calculated.

1.2.2.2 Free Hand-Eye Calibration

Another solution is Free hand-eye calibration [1] which is based on the Tsai's camera calibration technique [41]. It is assumed that the robot end effector moves in the plane parallel with 2D work space. The strategy solves the hand-eye calibration problem for robotic object tracking, without estimating the hand-eye transformation matrix. It first uses Tsai calibration technique to calibrate the camera. Tsai's calibration method uses a pinhole model of 3D-2D perspective projection with a first order radial lens distortion parameter. This camera model is described by five intrinsic and six extrinsic parameters. As the robot end effector moves from point A along only X and Y axis in the robot reference frame to point B, the coordinates of point B is obtained by applying the result of camera calibration and taking into account the moving of the end effector as an offset, $(X_B, Y_B) = (X_A, Y_A) + (X_e, Y_e)$, where (X_A, Y_A) and (X_B, Y_B) are the real world

coordinates of the points, and (X_e, Y_e) is the offset. This process is repeated after each robot end effector movement for object tracking and avoids computing the hand-eye relationship.

1.3 Motivation, Objective and Contribution

One of the most related areas of visual servoing is computer vision, which is defined in [57] as a set of computational techniques aimed at estimating or making explicit the geometric and dynamic properties of the 3-D world from digital images. According to this definition, the mapping from image to real world is the problem that needs to be solved first and foremost. It is well known that the mapping is nonlinear especially taking the image distortions into account. Tsai [41] and Zhang [47] develop nonlinear methods to model the camera. However, practically no perfect model is able to describe a vision system due to the assumption of derivation. Therefore, the mathematical model of camera could not always satisfy the high precision. On the other hand, although solving AX=XB and free hand-eye calibration give acceptable results, the tracking precision is still under improvement. Thus the first objective of this research is to design a new algorithm to increase the precision of hand-eye calibration to improve the accuracy of PBVS. It is noted that the mapping from object image coordinates to real world object coordinates, while the camera moves with the robot, is highly non-linear. Neural networks are the prime candidates to approximate nonlinear mapping due to their universal approximate ability.

The second objective is to improve the classical IBVS to achieve better performance of visual servoing. It is well known that traditional IBVS has some weakness including

image singularities, image local minima and relative long trajectory in Cartesian space. Recent research efforts on the problem have been thus concentrated on avoiding these disadvantages. The second part of this research is to develop a new visual servoing system to avoid the disadvantages of traditional IBVS and control the position of the end effector with respected to the static object as required by the task.

According to the above objectives, the research contributions including two novel methods for visual servoing in robotic manufactory systems are presented as follows.

1. A novel NN Based Hand-Eye Calibration is developed.
    - A MultiLayer Perceptron (MLP) NN is designed to approximate the highly nonlinear coordinate transform from object image coordinates to real world object coordinates as the camera moves with the robot.
    - The proposed method is simulated in Matlab/Simulink to track a moving object whose trajectory is a synthetic sinusoid in the 2D workspace.
    - Experiment of 2D object tracking is implemented, and the comparison with other existing methods is provided.
2. A new approach to IBVS with laser system is proposed.
    - A laser pointer is adopted to attain the depth by laser triangulation and is chosen as an image feature that is highly easy for image processing.
    - A three stages switching IBVS with laser controller is designed.
    - The proposed method is simulated in Matlab/Simulink by achieving four representative tasks. The analysis and comparison are provided.

- One of the applications of the IBVS with laser pointer system for robotic assembly is implemented in a robotic manufacturing system. Both good and bad calibration results are tested in the system to validate the practicability.

Publication related to the research is [56]

- Z. Li, W. F. Xie and N. Aouf, "A Neural Network Based Hand-Eye Calibration Approach in Robotic Manufacturing Systems", CSME 2006, Calgary, May 2006.

Papers intend to be submitted are

- "A New Approach to Switching Control of Image Based Visual Servoing with Laser Pointer," IEEE Transactions on Industrial Electronics.
- "A Novel Image Based Visual Servoing with Laser Pointer System," IEEE International Conference on Systems, Man, and Cybernetics, 2007

1.4 Thesis Organization

The dissertation is composed of six chapters. The introduction chapter introduces the basic concepts and relative background. Motivation, objective and contribution are presented as well.

Chapter 2 introduces a novel NN based hand-eye calibration method in robotic manufacturing system. A MLP NN is used to approximate the nonlinear coordinate transform from object image coordinates to real world object coordinates in robotic visual servoing.

Chapter 3 includes the simulation and experimental results of NN based hand-eye calibration. The analysis and comparison are given to validate the proposed method.

Chapter 4 introduces a new approach to switching control of IBVS with laser pointer.

The simple off the shelf laser pointer based triangulation method is applied to realize the depth estimation.

Chapter 5 includes the simulation and experimental results of IBVS with laser pointer system. The analysis and comparison are given to verify the effectiveness of the proposed method.

Chapter 6 contains the conclusion and a summary of the future work.

Appendix A gives the UPJ robot model in Matlab Robotics Toolbox.

Appendix B contains the pseudocode for NN training by Matlab Neural Network Toolbox.

Appendix C presents the user interface and the flow chart of teaching and assembly procedures.

1.5 Summary

In this chapter, basic concepts and components of visual servoing system are introduced. The background of visual servoing, including the classification, the advantages and disadvantages of each class, is addressed. An overview of hand-eye calibration is presented as well. The motivation, objective and contribution of this research have also been included to address the purpose and current achievements of the research.

CHAPTER 2 NN BASED HAND-EYE CALIBRATION

2.1 Introduction

In this chapter, a novel NN Based Hand-Eye Calibration is introduced in robotic manufacturing system to achieve a more accurate performance of the vision system. A MLP NN is used to approximate the nonlinear coordinate transform from object image coordinates to real world object coordinates in visual servoing. Based on the Levenberg-Marquardt (LM) optimization technique, the back-propagation algorithm is adopted to train NN. The inputs of the NN are the image coordinates and the robot end effector position with respect to the robot base. The outputs are the real world object coordinates.

2.2 Mapping from Real World Point to Image Point

The configuration for hand-eye calibration is shown in Figure 2-1, which is composed of a 6DOF robot, a camera mounted on the robot end effector and a calibration pattern in 2D workspace. It is assumed that there are only three translational motions of the robot end effector.

First the camera model is derived. The camera model is used to link the position of scene points with that of their corresponding image points [57]. The most common geometric model of an intensity camera is the perspective model. We define the camera focal length λ, the image center pixel coordinates (u_0, v_0), the lens distortion coefficient k_1, and the effective size of the pixel in the horizontal and vertical direction (s_x, s_y), as

intrinsic camera parameters, which link the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame.

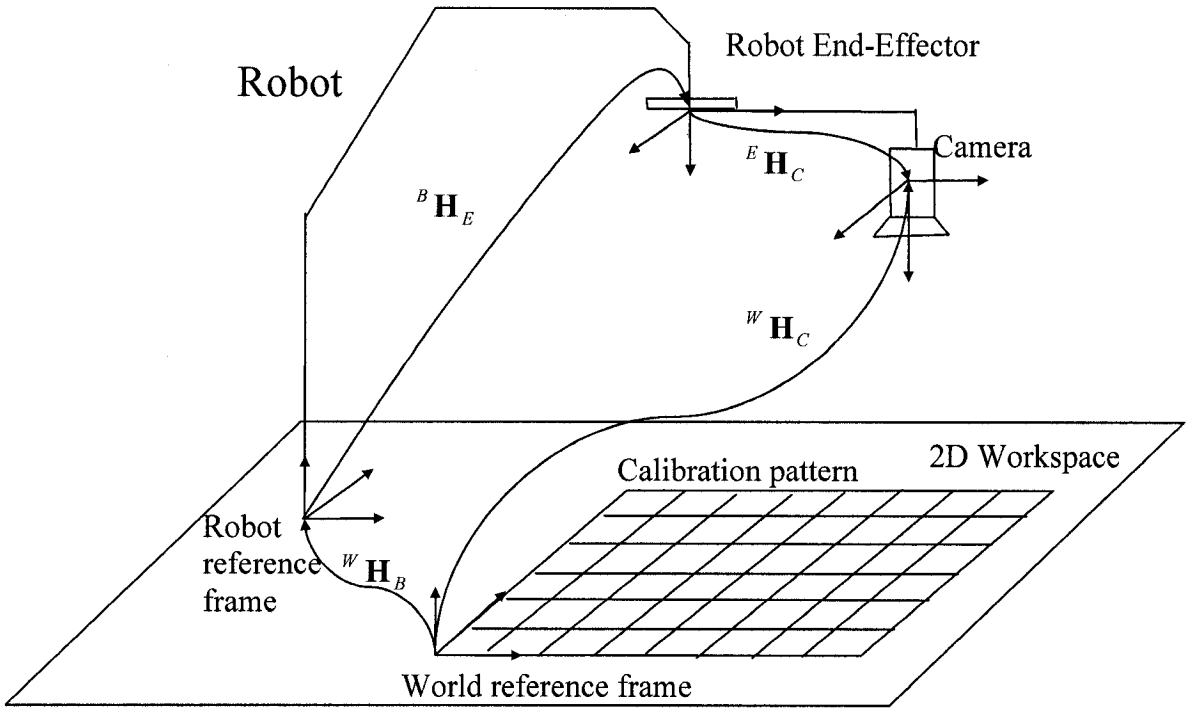


Figure 2-1 Robotic eye in hand system configuration

The perspective cameral model is shown in Figure 2-2. Referring to Figure 2-2 and using perspective projection with pinhole camera geometry, one can obtain the transformation from camera coordinates (x_c, y_c, z_c) to undistorted image coordinates (x_u, y_u, z_u):

$$x_u = \lambda \frac{x_c}{z_c} \tag{2-1}$$

$$y_u = \lambda \frac{y_c}{z_c} \tag{2-2}$$

$$z_u = \lambda \tag{2-3}$$

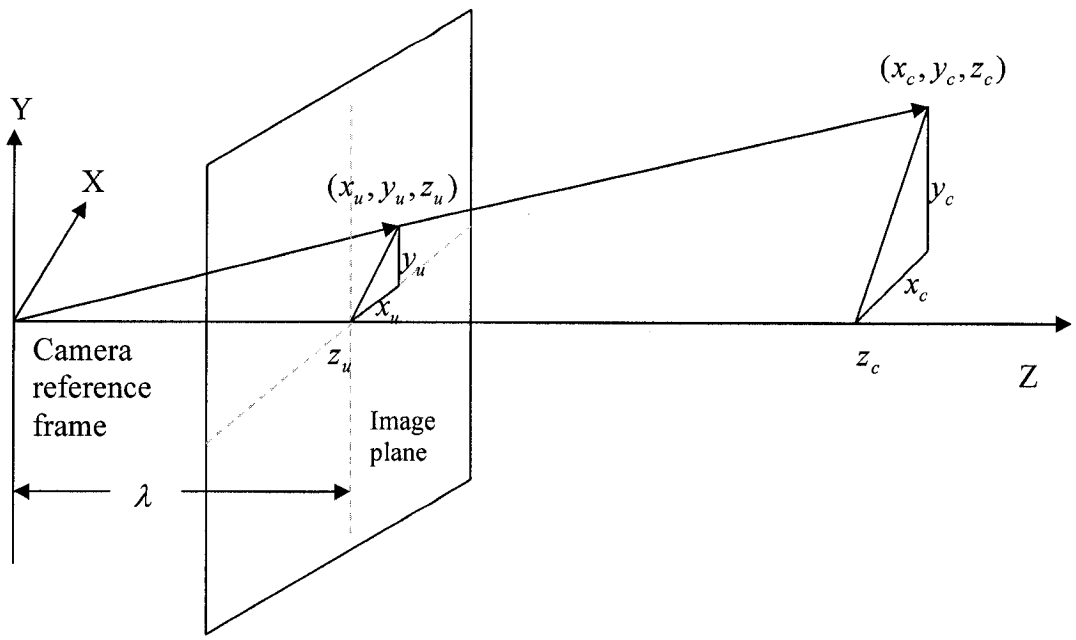


Figure 2-2 The perspective camera model

When the geometric distortion introduced by the camera is taken into account, the relationship between undistorted (x_u, y_u) and distorted image coordinates (x_d, y_d) is presented as:

$$x_u = x_d + k_1 x_d (x_d^2 + y_d^2) \tag{2-4}$$

$$y_u = y_d + k_1 y_d (x_d^2 + y_d^2) \tag{2-5}$$

The transformation between the pixel indices and the coordinates with respect to the camera reference frame is presented below:

$$x_d = -(u - u_0) s_x \tag{2-6}$$

$$y_d = -(v - v_0) s_y \tag{2-7}$$

The transformation between the camera reference frame and the image reference frame can be defined as

$$\mathbf{M}_{\text{int}} = \begin{bmatrix} -\frac{\lambda}{s_x} & 0 & u_0 \\ 0 & -\frac{\lambda}{s_y} & v_0 \end{bmatrix} \tag{2-8}$$

When the radial distortion is neglected, the transformation between pixel indices (u, v) and the coordinates with respect to the camera reference frame (x_c, y_c, z_c) is presented as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{M}_{\text{int}} \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \\ \\ 1 \end{bmatrix} \tag{2-9}$$

The relationship among the world reference frame, robot reference frame, robot end effector frame, and camera reference frame is shown as follows:

$${}^{W}\mathbf{H}_{C} = {}^{W}\mathbf{H}_{B}\,{}^{B}\mathbf{H}_{E}\,{}^{E}\mathbf{H}_{C} \tag{2-10}$$

where ${}^{W}\mathbf{H}_{B}$ and ${}^{E}\mathbf{H}_{C}$ are fixed during the robot end effector moving.

A sequence of three rotations roll, pitch, yaw (ϕ, θ, ψ) and three translations (T_x, T_y, T_z) are used to represent ${}^{B}\mathbf{H}_{E}$ as follows:

$${}^{B}\mathbf{H}_{E} = \begin{bmatrix} {}^{B}\mathbf{R}_{E} & {}^{B}\mathbf{T}_{E} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2-11}$$

in which

$${}^{B}\mathbf{T}_{E} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^{T} \tag{2-12}$$

$${}^{B}\mathbf{R}_{E} = \begin{bmatrix} \cos(\phi)\cos(\theta) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \sin(\phi)\cos(\theta) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) \\ -\sin(\theta) & \cos(\theta)\sin(\psi) & \cos(\theta)\cos(\psi) \end{bmatrix} \tag{2-13}$$

where ${}^{B}\mathbf{T}_{E}$ is the three translational motions of the robot end effector and ${}^{B}\mathbf{R}_{E}$ is the rotation matrix neglected in the mapping.

Thus (2-10) can be rewritten as

$$ {}^{W}\mathbf{H}_{C} = {}^{W}\mathbf{H}_{B} \begin{bmatrix} {}^{B}\mathbf{R}_{E} & {}^{B}\mathbf{T}_{E} \\ \mathbf{0} & 1 \end{bmatrix} {}^{E}\mathbf{H}_{C} = \begin{bmatrix} {}^{W}\mathbf{R}_{C} & {}^{W}\mathbf{T}_{C} \\ 0 & 1 \end{bmatrix} \tag{2-14} $$

where ${}^{W}\mathbf{H}_{B}$ and ${}^{E}\mathbf{H}_{C}$ are unknown.

The transformation ${}^{W}\mathbf{H}_{C}$ that consists of the rotation matrix

$$ {}^{W}\mathbf{R}_{C} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2-15} $$

and the translation vector

$$ {}^{W}\mathbf{T}_{C} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^{T} \tag{2-16} $$

are represented by the extrinsic camera parameters $[t_x \quad t_y \quad t_z \quad \phi_z \quad \phi_y \quad \phi_x]$, which define the translations and orientations of the camera reference frame with respect to the world reference frame.

From (2-1) to (2-14), we have the mapping from a 3D point (x_w, y_w, z_w) in the world reference frame to the camera reference frame

$$ x_u = \lambda \frac{r_{11}x_W + r_{12}y_W + r_{13}z_W + t_x}{r_{31}x_W + r_{32}y_W + r_{33}z_W + t_z} \tag{2-17} $$

$$ y_u = \lambda \frac{r_{21}x_W + r_{22}y_W + r_{23}z_W + t_y}{r_{31}x_W + r_{32}y_W + r_{33}z_W + t_z} \tag{2-18} $$

When the camera radial distortion is introduced, substituting (2-6) and (2-7) into (2-4) and (2-5) yields

$$x_u = -(u-u_0)s_x - k_1(u-u_0)s_x[(u-u_0)^2 s_x^2 + (v-v_0)^2 s_y^2] \tag{2-19}$$

$$y_u = -(v-v_0)s_y - k_1(v-v_0)s_y[(u-u_0)^2 s_x^2 + (v-v_0)^2 s_y^2] \tag{2-20}$$

Then substituting (2-19) and (2-20) into (2-17) and (2-18), we have

$$\lambda \frac{r_{11}x_W + r_{12}y_W + r_{13}z_W + t_x}{r_{31}x_W + r_{32}y_W + r_{33}z_W + t_z} = -(u-u_0)s_x - k_1(u-u_0)s_x[(u-u_0)^2 s_x^2 + (v-v_0)^2 s_y^2] \tag{2-21}$$

$$\lambda \frac{r_{21}x_W + r_{22}y_W + r_{23}z_W + t_y}{r_{31}x_W + r_{32}y_W + r_{33}z_W + t_z} = -(v-v_0)s_y - k_1(v-v_0)s_y[(u-u_0)^2 s_x^2 + (v-v_0)^2 s_y^2] \tag{2-22}$$

Equations (2-21) and (2-22) describe the mapping function, from (T_x, T_y, T_z) in the robot reference frame, the depth z_W, and the object image pixel coordinates (u, v) to real world object coordinates (x_W, y_W). Define the mapping function f as follows,

$$\begin{bmatrix} x_W \\ y_W \end{bmatrix} = f(T_x, T_y, T_z, u, v, z_W) \tag{2-23}$$

It is noted that the mapping from object image coordinates to real world object coordinates, while the camera moves with the robot, is highly non-linear. Neural networks are the prime candidates to approximate such nonlinear mapping due to their universal approximate ability.

2.3 NN Based Hand-Eye Calibration

According to the derivation of mapping from real world point to image point when camera is mounted on the robot end effector, a new hand-eye calibration method is proposed to solve the hand-eye calibration problem using a neural network. This

approach avoids the calculation of geometrical transformations, such as solving AX=XB equation.

2.3.1 NN Architecture

For a fixed camera position, a neural network structure, with inputs as the image coordinates and outputs as the real world coordinates, can solve the camera calibration problem. However, for a camera mounted on the robot end effector, this neural network structure is not able to predict accurately the object real world coordinates because the camera position changes with the robot motion. Thus, besides the image coordinates, the robot end effector position with respect to the robot base must be taken into consideration to solve mapping problem.

A MLP NN shown in Figure 2-3 is used to approximate the nonlinear coordinate transform. The MLP structure is composed of one input layer, one hidden layer and one output layer. The inputs of the network are the image coordinates $(u, v) = (x_{io}, y_{io})$ and the robot end effector position with respect to the robot base $(T_x, T_y, T_z) = (x_e, y_e, z_e)$. The outputs of the network are the real world coordinates (X_0, Y_0). The activation function used for the hidden layer nodes is hyperbolic tangent sigmoid:

$$f_1(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \tag{2-24}$$

where n is the net input.

The activation function used for the output layer nodes is linear transfer function:

$$f_2(n) = n \tag{2-25}$$

The number of nodes for hidden layer is fixed at twenty, which is obtained by trial and error.

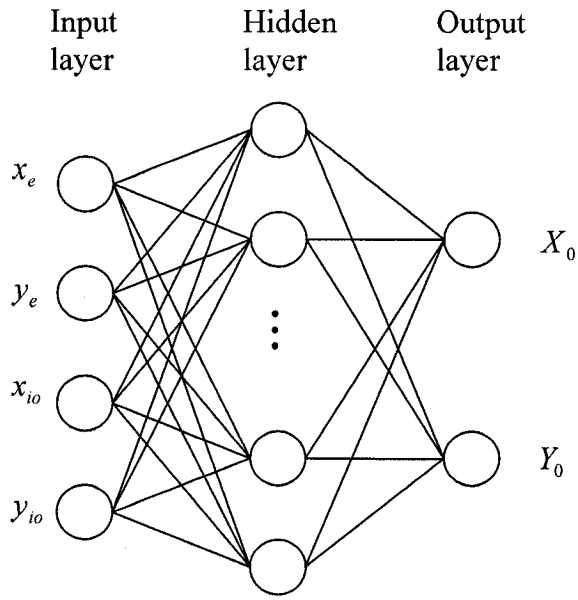


Figure 2-3 Distributed NN structure

Multiple viewpoints are generated at the same heights in the world reference frame (M_v viewpoints in total) by moving the camera, and then an image of the planar calibration pattern is taken at each position. The pattern comprises N_c corners of black and white squares. Based on (2-21) and (2-22), to minimize the world coordinates error between the measurement coordinates $\{(X_0)_j^i,(Y_0)_j^i\}$ and the outputs of the NN $\{(\hat{X}_0)_j^i,(\hat{Y}_0)_j^i\}$, the objective function is defined as:

$$\min E = \sum_{i,j}^{M_v,N_i} \left\{ \left((X_0)_j^i - (\hat{X}_0)_j^i\right)^2 + \left((Y_0)_j^i - (\hat{Y}_0)_j^i\right)^2 \right\} \tag{2-26}$$

i=1, 2, 3..., M_v

j=1, 2, 3..., N^i

where N^i denotes the number of corners in one viewpoint. Therefore, the total number of training data pairs is

$$N_a = \sum_{i=1}^{M_v} N^i \tag{2-27}$$

The NN training algorithm is chosen as back-propagation, based on the Levenberg-Marquardt (LM) optimization algorithm. The algorithm is a variation of Newton's method. It was designed to minimize functions that are sums of squares and fit for neural network training for which the performance index is the mean squared error [58]. This algorithm is designed to approach second-order training speed without having to compute the Hessian matrix.

Define N_a and M as number of input/target pair and number of NN output respectively. One can convert (2-26) into matrix form as:

$$\min E = \sum_{q=1}^{N_a} \mathbf{e}_q^T \mathbf{e}_q = \sum_{q=1}^{N_a} \sum_{j=1}^{M} (e_{j,q})^2 = \sum_{i=1}^{N_a \times M} (v_i)^2 \tag{2-28}$$

where $e_{j,q}$ is the j^{th} element of the error for the q^{th} input/target pair, and v_i is the i^{th} element of the error.

Let $N = N_a \times M$, the above equation can be rewritten as:

$$E(\mathbf{x}) = \sum_{i=1}^{N} v_i^2(\mathbf{x}) \tag{2-29}$$

which is a sum of squares function and is to be minimized with respect to the parameter vector x. According to this performance index, Newton's method would be

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\nabla^2 F(\mathbf{x})\right)^{-1} \nabla F(\mathbf{x}) \tag{2-30}$$

then the jth element of the gradient would be

$$[\nabla F(\mathbf{x})]_j = \frac{\partial F(\mathbf{x})}{\partial x_j} = 2\sum_{i=1}^{N} v_i(\mathbf{x}) \frac{\partial v_i(\mathbf{x})}{\partial x_j} \tag{2-31}$$

The gradient can therefore be written in matrix form:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}) \tag{2-32}$$

where

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{bmatrix} \tag{2-33}$$

is the Jacobian matrix.

The k, j element of the Hessian matrix is

$$[\nabla^2 F(\mathbf{x})]_{k,j} = \frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_j} = 2\sum_{i=1}^{N}\left\{\frac{\partial v_i(\mathbf{x})}{\partial x_k}\frac{\partial v_i(\mathbf{x})}{\partial x_j} + v_i(\mathbf{x})\frac{\partial^2 v_i(\mathbf{x})}{\partial x_k \partial x_j}\right\} \tag{2-34}$$

which is expressed in matrix form:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}) \tag{2-35}$$

where

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^{N} v_i(\mathbf{x})\nabla^2 v_i(\mathbf{x}) \tag{2-36}$$

$\mathbf{S}(\mathbf{x})$ is small enough to be neglected. Then the Hessian matrix is approximated as

$$\nabla^2 F(\mathbf{x}) \cong 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) \tag{2-37}$$

Substituting (2-32) and (2-37) into (2-30), one obtains the Gauss-Newton method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \tag{2-38}$$

The LM algorithm is presented as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \tag{2-39}$$

which also can be presented as:

$$\Delta \mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \tag{2-40}$$

While the scalar μ_k is increased, the above equation approaches:

$$\mathbf{x}_{k+1} \cong \mathbf{x}_k - \frac{1}{\mu_k}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) = \mathbf{x}_k - \frac{1}{2\mu_k}\nabla F(\mathbf{x}) \tag{2-41}$$

which is the steepest descent algorithm with small learning rate.

While μ_k is decreased to zero, the algorithm becomes Gauss-Newton method [58][59].

2.3.2 NN Based Hand-Eye Calibration Procedures

In this strategy, we consider a simple form involves robot end effector motion in a 2D plane parallel with 2D workspace.The following three calibration steps are taken.

Step1: Data collection.

The training set of the NN is attained by extracting a series of easily detectable features and their corresponding precise positions in world reference frame. In addition, the robot end effector positions can be obtained from robot controller. There are many kinds of targets for camera calibration, such as single plane of square boxes, plane of circles, orthogonal planes of square boxes and so on. A calibration pattern board with black and white checks is chosen, partially shown in Figure 2-4. The major advantage of this type of pattern is that the corners of each black and white check are easily to be extracted by corner detection.

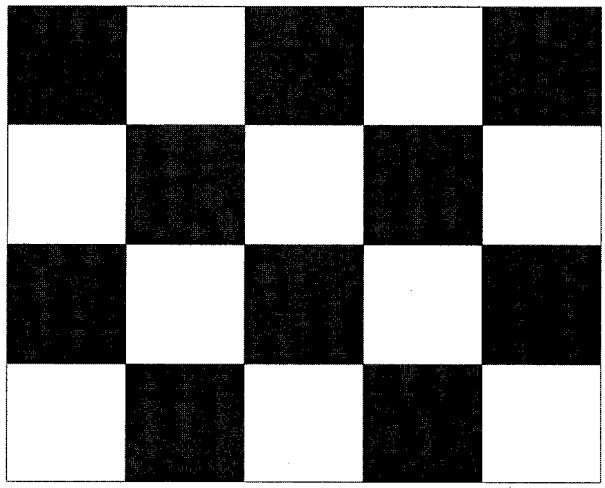


Figure 2-4 Calibration pattern

Locate the calibration pattern board shown in Figure 2-4 on the table and define the world reference frame coincided with robot reference frame. Since the robot end effector moves in the plane parallel with 2D workspace, we can thereafter move the camera just along X and Y axis in robot reference frame to different viewpoints. This procedure is demonstrated in Figure 2-5.

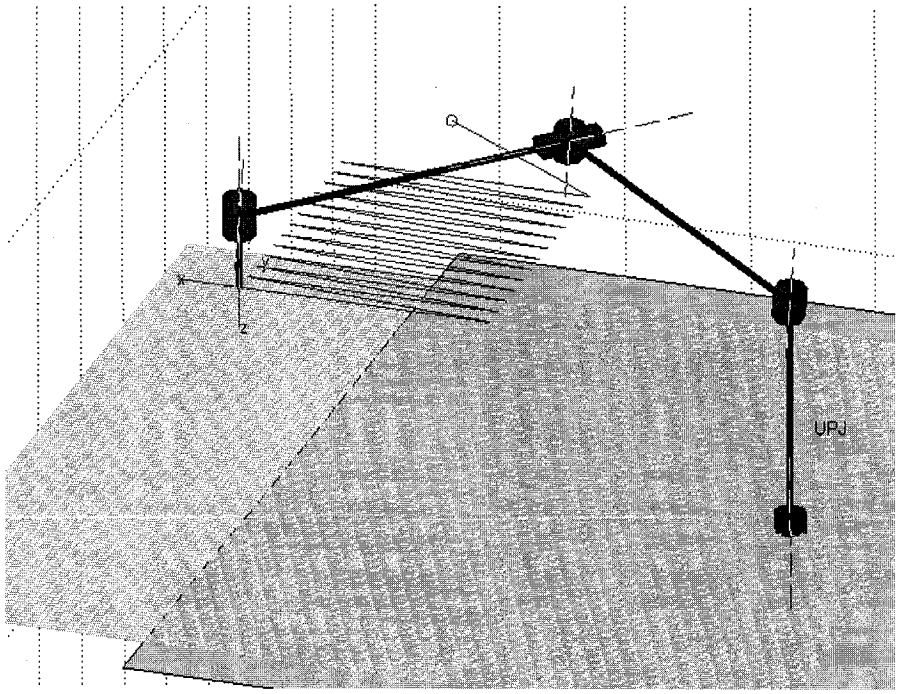


Figure 2-5 Robot end effector trajectory of all the view points

For each viewpoint, image of the calibration pattern are taken. The workspace plane is partitioned, for training data acquisition, into discrete locations to which the robot moves. For each robot position, a large number pairs of image coordinates points and real world coordinates points are selected. The collected data has to be dense and cover the workspace in which the object is moving.

Step2: NN Training.

The training performance of the neural network is greatly improved by normalizing the data outputs as:

$$(X_0, Y_0) = (X_0 / \max(X_0), Y_0 / \max(Y_0)) \tag{2-42}$$

where $\max(X_0)$ and $\max(Y_0)$ represent the limits of the workspace real world coordinates.

The performance index used for MLP training is the mean squared error. If each target occurs with equal probability, the mean squared error will be proportional to the sum squared error over all targets in the training set, which is presented in (2-29). The Levenberg Marquardt iterations composed of four steps are presented below [58]:

(i) Introduce all inputs to the neural network and compute the corresponding outputs and the errors. Then compute the sum squared errors for all inputs using (2-29).

(ii) Compute the Jacobian matrix by (2-33). Calculate the elements of the Jacobian matrix.

(iii) Solve (2-40) to obtain $\Delta \mathbf{x}_k$

(iv) Recalculate the sum squared errors using $\mathbf{x}_k + \Delta \mathbf{x}_k$. If the new value is smaller than the value in Step(i), then divide μ by a factor ϑ. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k$ and

go back to Step(i). In contrast, if the sum squared error is not reduced, multiply μ by ϑ and go back to Step(iii).

Step3: NN Generalization

One of the major advantages of neural networks is their ability of generalization. Generalization of NN stands for the ability to have the outputs of the network when the inputs are not in the training set. In our case, once the NN has been trained, it serves as real world coordinates predicator for each robot position within the training area.

The NN generalization is carried out by inputting a number of image coordinates and relevant robot end effector positions and checking the NN outputs, real world object coordinates.

Once the object is detected, the neural network predicts the real world object coordinates, which are sent to the robot for tracking. Before being sent to the robot, the real world object coordinates obtained from the neural network are scaled back as follows:

$$(X_0, Y_0) = (X_0 \cdot \max(X_0), Y_0 \cdot \max(Y_0)) \tag{2-43}$$

2.4 PBVS Using NN Based Hand-Eye Calibration

We consider a simple form of visual servoing, which involves robot end effector motion in a plane parallel with 2D workspace. This can be used for tracking planar motion such as a conveyor belt. We begin by considering point to point positioning task, which indicates bringing a point ${}^{E}\mathbf{P}$ rigid linking the robot to a stationary point **S** in the workspace [4].

The control diagram is presented in Figure 2-6.

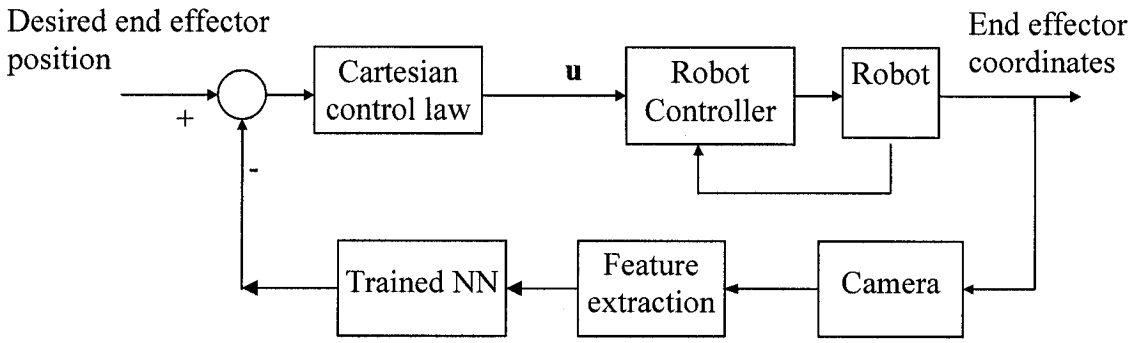


Figure 2-6 Control diagram of PBVS using NN based hand-eye calibration

Since the robot end effector has three translational DOF, the control input to be computed is the desired end effector translational velocity, as denoted by **u**. Normally, by using traditional hand-eye calibration, the kinematic error function can be defined in end effector frame as

$$^{E}\mathbf{e}=^{E}\mathbf{P}-^{E}\mathbf{S} \tag{2-44}$$

Thus the proportional control law can be express as:

$$\mathbf{u}=-K^{E}\mathbf{e}=-K(^{E}\mathbf{P}-^{E}\mathbf{H}_{C}{}^{C}\mathbf{S}) \tag{2-45}$$

where K is the proportional gain and $^{C}\mathbf{S}$ is point **S** with respect to camera reference frame.

By applying NN based hand-eye calibration, one chooses $^{E}\mathbf{P}$ as the projecting point of the end effector on the 2D workspace, $^{W}\mathbf{P}$. Therefore, the kinematic error function defined in end effector frame can be transformed to world reference frame as:

$$^{E}\mathbf{e}\ =^{W}\mathbf{e}\ =^{W}\mathbf{P}-^{W}\mathbf{S} \tag{2-46}$$

where $^{W}\mathbf{S}$ is the real world coordinates that have been achieved from the trained NN. The proportional control law may be express as:

$$\mathbf{u} = -K^{W}\mathbf{e} = -K({}^{W}\mathbf{P} - {}^{W}\mathbf{S}) \tag{2-47}$$

The control is achieved by a sequence of independent steps. Firstly, a desired end effector position is generated as a reference signal. Secondly, the position of the end effector with respect to the fixed point in the workspace is estimated. The third step is to use the Cartesian control law to calculate the Cartesian motion required to move the end effector to the desired position. Finally the robot is moved by the calculated control signal. These steps will be repeated until the end effector is at the desired position.

2.5 Limitation

It is assumed that the robot end effector moves just in a plane parallel with 2D workspace. When the object has 6DOF pose with respect to the camera or the robot end effector moves with 6DOF, the NN based hand-eye calibration will become ineffective.

As discussed previously, accurate hand-eye calibration is the foundation of PBVS. However, PBVS has inherent limitations. For example, when the object model is not available, PBVS method does not work any more. This limitation restricts many applications of visual servoing. On the contrary, considering one of the advantages of IBVS is object model free, we will develop a novel IBVS in the second part of the research.

2.6 Summary

In this chapter, a novel NN Based Hand-Eye Calibration is introduced in robotic manufacturing system. A MLP NN is used to approximate the nonlinear coordinate transform from object image coordinates to real world object coordinates in robotic visual servoing. Calibration steps are presented in detail and point to point PBVS using NN

based hand-eye calibration method is briefly introduced. Finally the limitations of the proposed algorithm are given as well.

CHAPTER 3 SIMULATION AND EXPERIMENTAL RESULTS OF NN BASED HAND-EYE CALIBRATION

3.1. Introduction

In this chapter, simulation and experiment are carried out to evaluate the performance of NN based hand-eye calibration.

Simulation results are given in the first part. The proposed method is trying to control the robot end effector to track a moving object whose trajectory is a synthetic sinusoid in 2D workspace. The error analysis is given subsequently.

In the experiment, some random points in a 2D workspace are chosen to simulate a moving object. The proposed method is used to detect the position of these points and track the virtual path. The error analysis and comparison with other methods are presented in the end.

3.2. Simulation Results of NN Based Hand-Eye Calibration

The proposed algorithm is simulated in Matlab Neural Network Toolbox [72] and the Robotics Toolbox [69]. The simulation is based on the robot model of Motoman UPJ, the camera model of Sony XC55 and a pattern of black and white checks whose side lengths are equal to 10mm. The robot model is presented in Appendix A and the camera model is given in Table 3-1.

All the viewpoints demonstrated in Figure 2-5 is generated by fixing $Z = 300mm$ and moving the robot end effector along X and Y axis at intervals of 20mm. The 2D workplace is in the range of 700×700 mm. The camera reference frame is assumed coincident with robot end effector frame.

Table 3-1 Camera parameters

Focal length	6mm
Effective size	$0.0074 \times 0.0074mm$
Image size	640×480 pixel
Principal point	[320 240] pixel

The training is carried out by Matlab Neural Networks Toolbox. The performance function is Mean squared error, the number of hidden layer nodes is fixed at twenty, the learning rate is 0.03, and the performance goal is 10^{-8}. The training curve is shown in Figure 3-1.

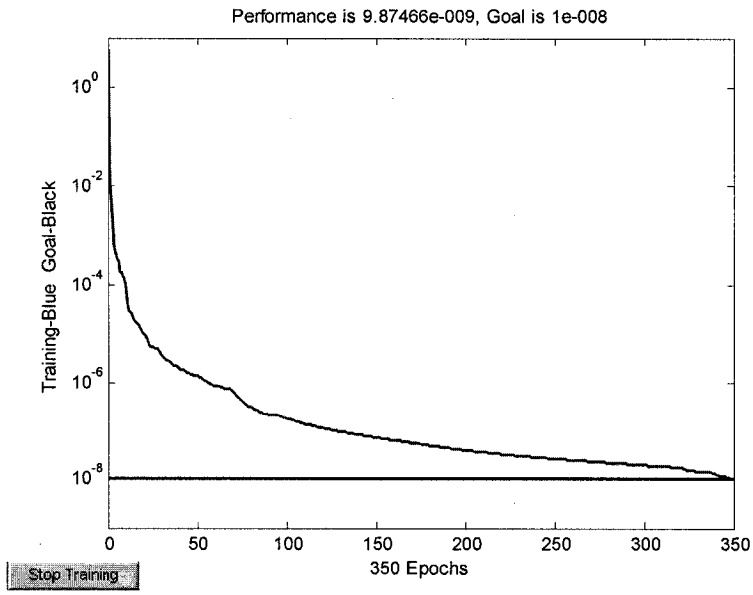


Figure 3-1 Training curve

To evaluate the performance, a synthetic sinusoid is to simulate a moving object in 2D workspace, whose equation is given by:

$$\begin{aligned} &Y = 100\sin(10\pi X) \quad (mm) \\ &X \in [200 \quad 400] \quad (mm) \\ &Z = 0 \end{aligned} \tag{3-1}$$

in which the step of X is 5mm.

The Cartesian trajectory for tracking the sinusoid is shown in Figure 3-2, where "o" and "*" denote initial position and final position respectively.

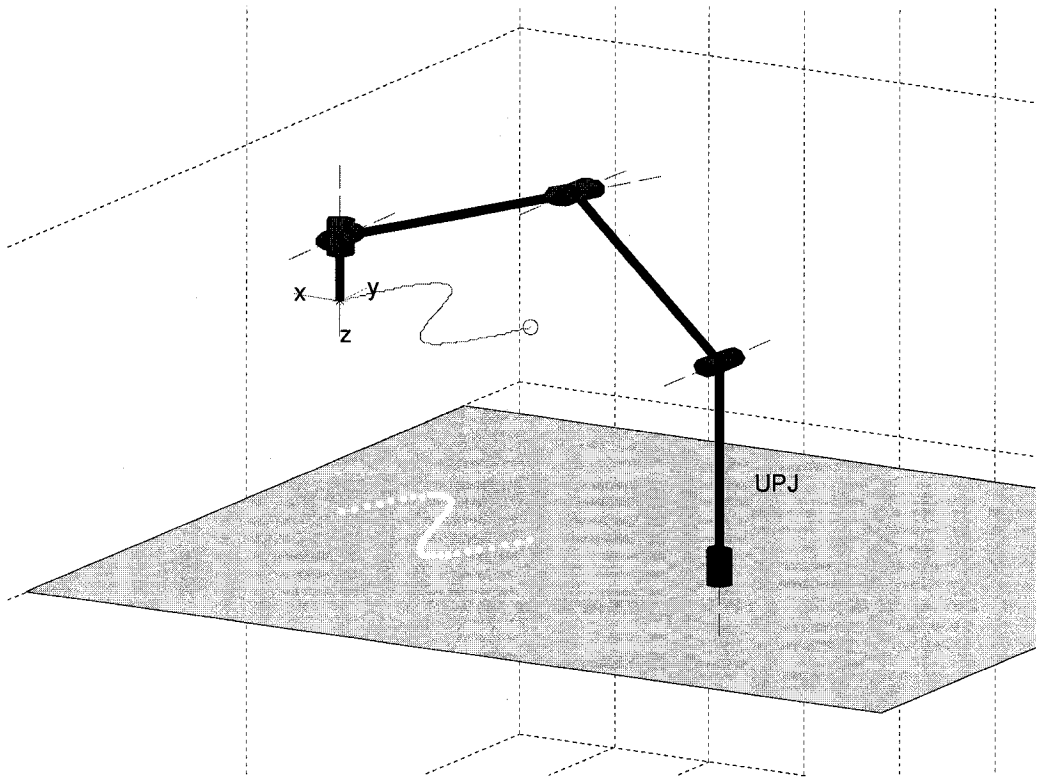


Figure 3-2 Cartesian trajectory for object tracking

The X-Y projection of 3D Cartesian trajectory shown in Figure 3-2 is represented in Figure 3-3, where all "·" are synthetic target points composing the sinusoid.

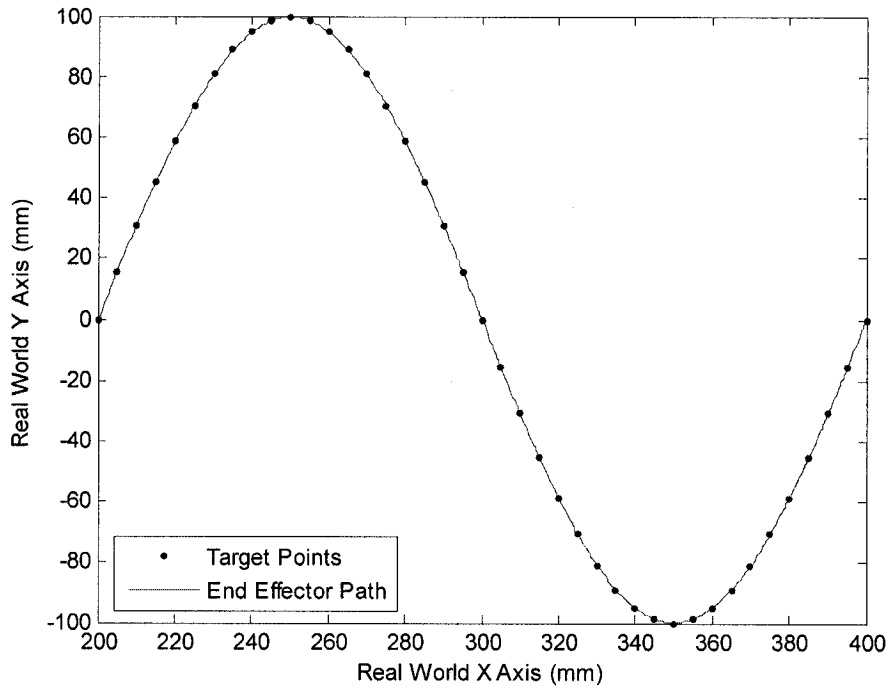


Figure 3-3 X-Y projection of 3D Cartesian trajectory

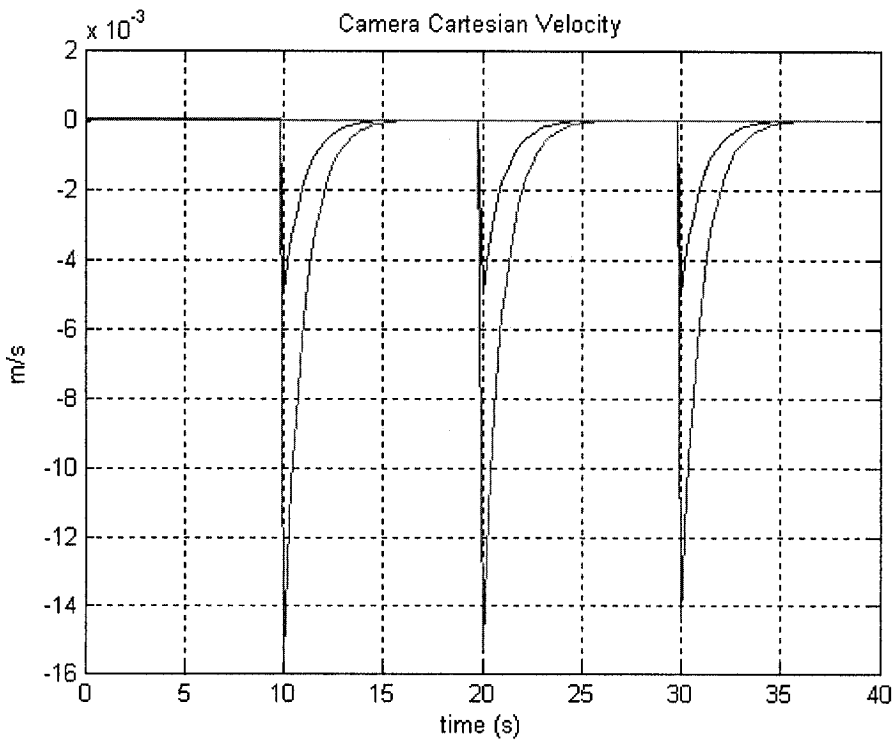


Figure 3-4 Camera Cartesian velocity for tracking first three points

Totally 41 points are selected for tracking. Figure 3-4 presents the 2DOF camera Cartesian velocity for tracking first three points. The time for tracking each point is 10 seconds. Table 3-2 gives the tracking error of NN based hand-eye calibration method. It is noticed that very accurate results, which are under 0.01mm, have been achieved by the proposed method.

Table 3-2 Tracking error of NN based calibration method

Method	Mean Error (mm)		
	X	Y	$\sqrt{X^2+Y^2}$
NN based hand-eye calibration	0.0048	0.0098	0.0050

3.3 Experimental System Introduction

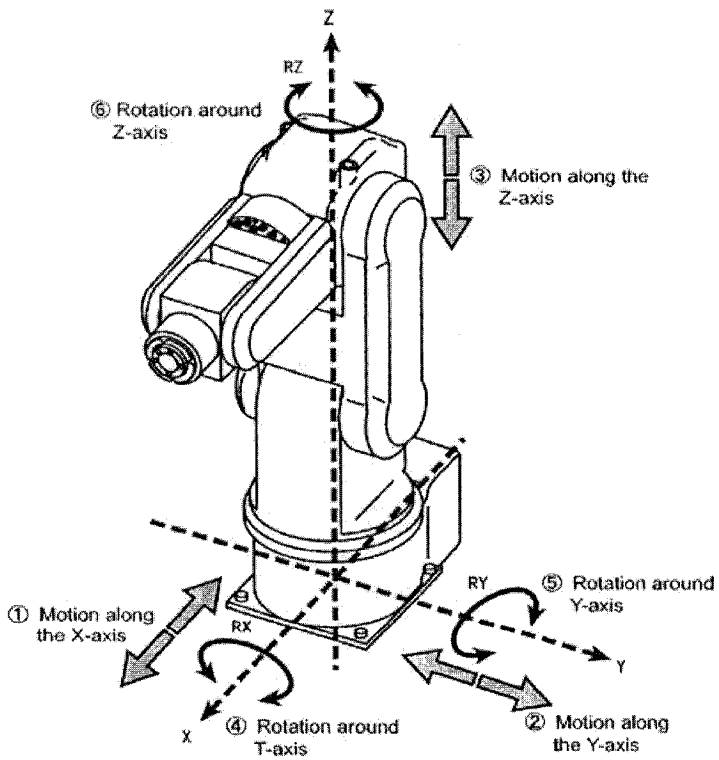


Figure 3-5 Coordinate frame for Motoman UPJ

The definition of coordinate frame for 6DOF Motoman UPJ robot manipulator is presented in Figure 3-5. The detailed experimental setup is shown in Table 3-3.

Table 3-3 Experimental system

Robot manipulator	Motoman UPJ (6DOF) A compact, high-speed robot that requires minimal installation space and offers a range of payloads for handling the smallest items [7]
Robot controller	JRC This controller offers flexibility and high performance, and is ideal for robotic applications such as small part handling and assembly [7]
Camera	Sony XC55 Monochrome CCD camera, which uses square pixels most suitable for machine vision [78].
Frame grabber	Matrox Meteor-II/Multi-Channel PCI form factor. Can capture from standard analog monochrome camera and transfer acquired images to PC memory for processing [79]
Camera adaptor	Sony DC-700 A small, lightweight camera adaptor integrated power supply and compatible with Sony XC-Series camera [83]

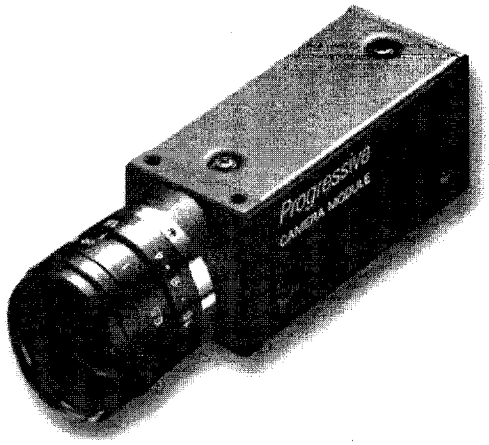


Figure 3-6 Sony XC55 monochrome CCD camera

The picture of used camera-- Sony XC55 monochrome CCD camera is demonstrated in Figure 3-6. The diagram of the experimental system is described in Figure 3-7, which is a typical look-and-move based architecture. Through the camera adaptor and frame grabber, the image is transformed into computer as vision feedback. Via image processing and the control law, the output of the visual servo controller is calculated and sent to the robot controller, which is a low level joint controller.

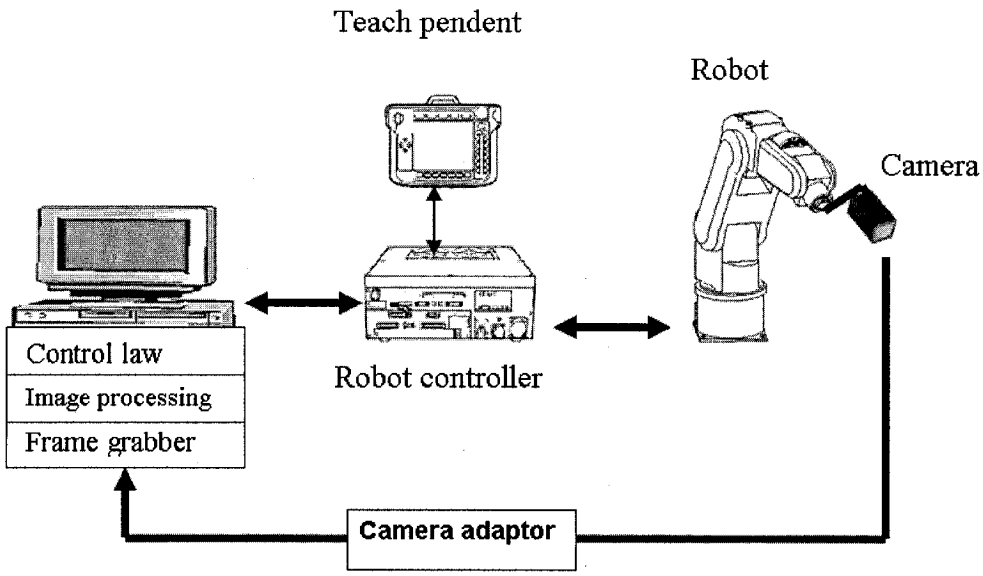


Figure 3-7 Schematic representation of the experimental system

3.4 Experimental Results of NN Based Hand-Eye Calibration

Partial calibration pattern is shown in Figure 3-8. Because of the lens distortation, a straight line in the real world becomes a curve in the image plane.

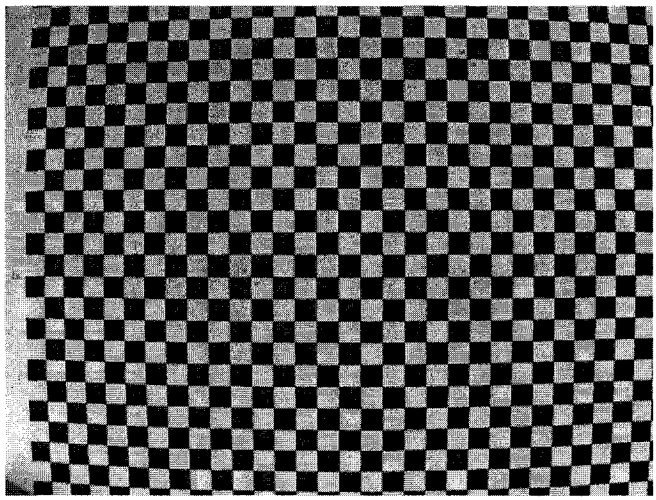


Figure 3-8 Calibration pattern in the image

The NN based hand-eye calibration algorithm has been tested on an experimental system shown in Figure 3-9.

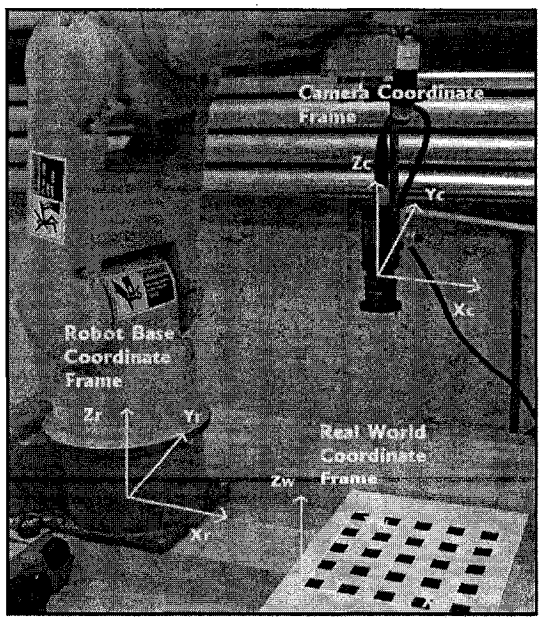


Figure 3-9 Robotic manufacturing system setup for hand-eye calibration

Referring to Figure 3-10, ten testing positions of robot end effector for generalization are chosen. The number of pairs $(x_e, y_e, x_{io}, y_{io}), (X_o, Y_o)$ used for training is approximately 16000.

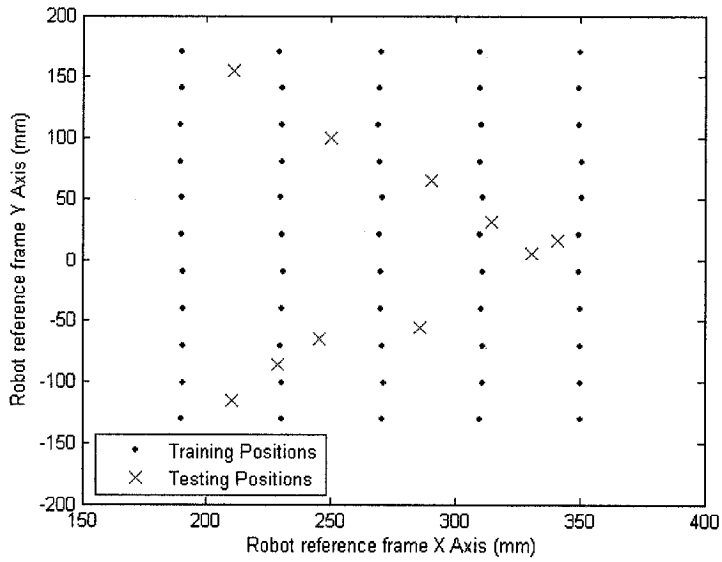


Figure 3-10 Training and testing positions of robot end effector

As shown in Figure 3-11, which is part of the calibration pattern, the six crosses are random actual real world coordinates, and the four spots around each cross denote the calculated real world coordinates from four robot end effector positions. For each position, the camera can only see partial calibration pattern, thus there are four spots instead of ten. The test result indicates that when camera located at different position gazes at same point, the calculated real world coordinates of the point all approximate to the actual coordinates.

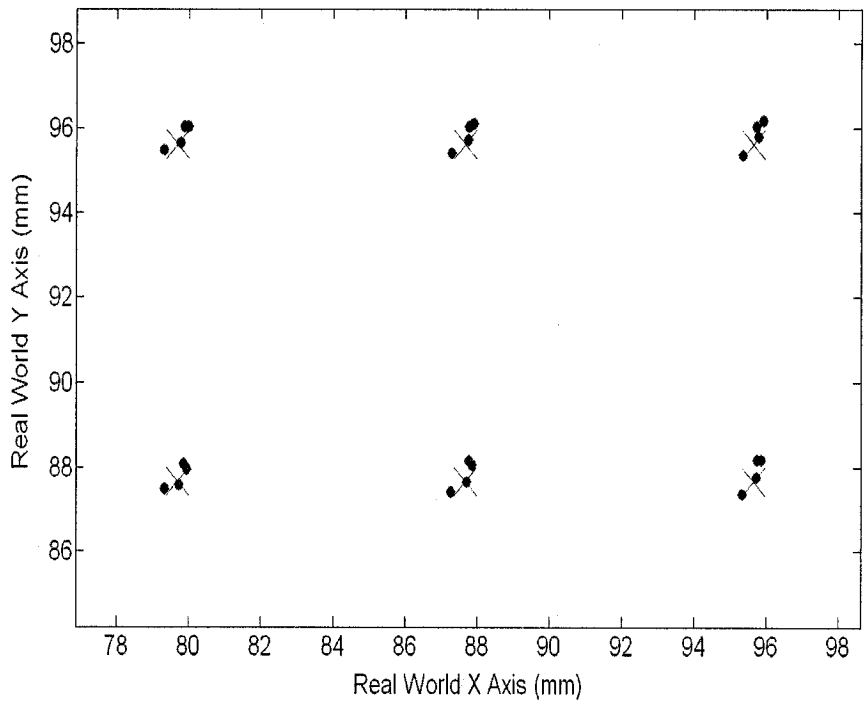


Figure 3-11 Neural Network test result

In order to test the calibration result, we simulate object tracking, which aims at detecting the position of a moving object from a video sequence.

First the path of object is planned by choosing some random points in the 2D workspace to simulate a moving object, and then the robot end effector is set to the initial position where the robot end effector is perpendicular to the 2D workspace and directly above the first point. After taking a picture for those points, we acquire pixels coordinates of the second point (x_{io}, y_{io}) from the image, and robot end effector position (x_e, y_e) relative to the robot base from the robot controller. Based on pixels coordinates and end effector position, the trained NN gives the real world coordinates (X_0, Y_0) of the second point. Using the difference value between the second real world coordinate and the first one as an offset, one can obtain the new robot end effector coordinate. Finally we move

the robot end effector along only X and Y axis in robot reference frame directly above the calculated position of the second point. This procedure is repeated for the chosen points one by one in the planned path. The robot end effector is moved along the planned path as the object tracking.

In Figure 3-12 to Figure 3-15, the spots represent the chosen points and the plus signs represent the robot end effector positions after tracking the target points, which can be regarded as the projection in the world reference frame.

Figure 3-12 and Figure 3-13 represent the experimental error analysis and the tracking path when using neural network strategy for object tracking. The calculated end effector path is almost coincident with the desired target path.

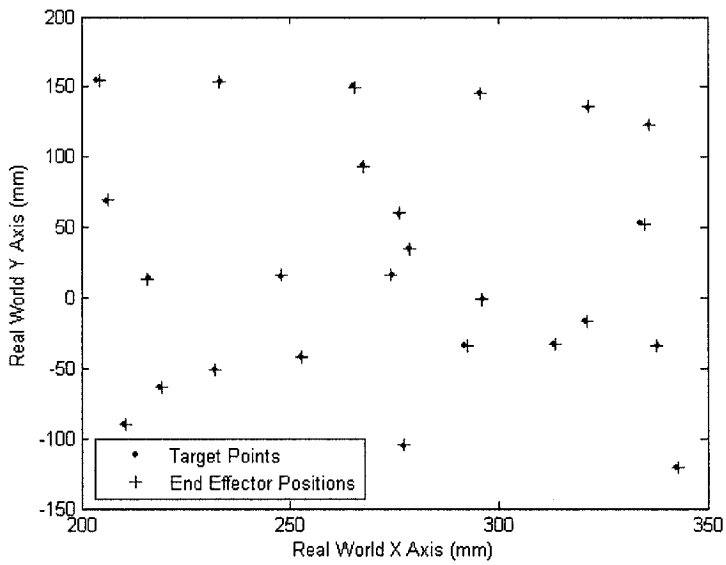


Figure 3-12 Desired points and calculated points by NN based hand-eye calibration

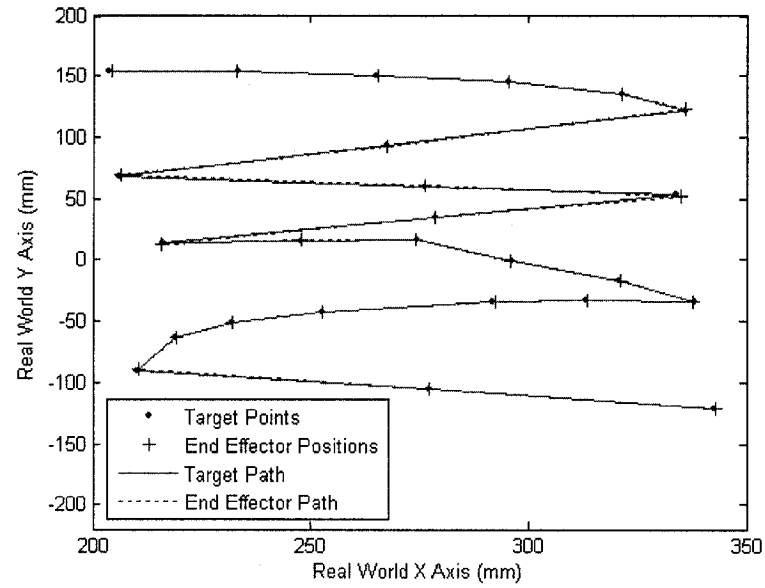


Figure 3-13 Tacking path by NN based hand-eye calibration

The proposed method is compared with free hand-eye calibration and Motai's hand-eye calibration methods. The performances of those two methods are evaluated by tracking the testing points as well.

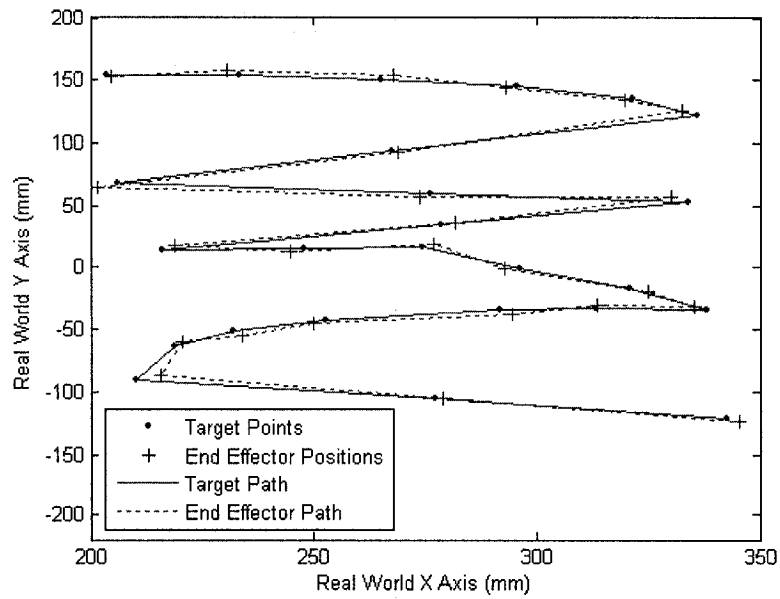


Figure 3-14 Tacking path by free hand-eye calibration

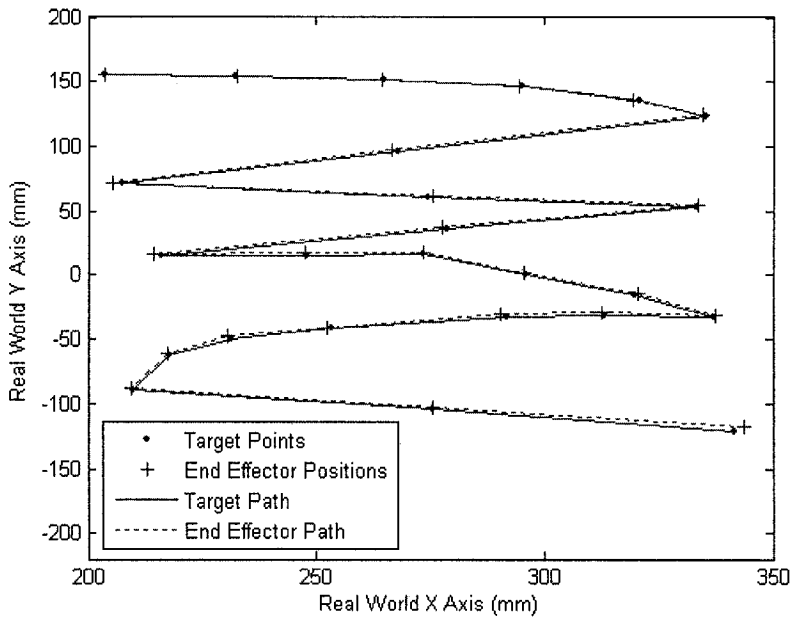


Figure 3-15 Tacking path by Motai's hand-eye calibration

From Figure 3-14 and Figure 3-15, it can be found that there are certain extent errors between the desired target paths and the calculated ones.

Table 3-4 Error comparison of three types of calibration methods

Method	Mean Error (mm)		
	X	Y	$\sqrt{X^2+Y^2}$
Free Hand-Eye Calibration	2.70	2.46	3.79
Motai Hand-Eye Calibration	0.78	1.45	1.73
NN based Hand-Eye Calibration	0.28	0.43	0.58

Table 3-4 summarizes the errors for all the testing points of different methods. From experimental results, it is noted that the proposed neural network strategy gives the best results comparing with the methods of free hand-eye calibration and Motai's hand-eye calibration for 2D object tracking. Positioning errors are reduced to less than one

millimeter of absolute precision for moving object tracking. The visual positioning algorithm proposed is of very good quality.

3.5 Summary

In this chapter, NN based hand-eye calibration has been simulated using Matlab Toolbox and experimentally tested on robotic manufacturing system.

By using position based visual servoing and tracking a synthetic sinusoid in 2D workspace for simulation, NN based hand-eye calibration method present an accurate result, whose mean tracking error is under 0.01mm.

The experimental results show that its precision of 2D tracking outperforms the current solving homogeneous transformation matrix method and free hand-eye calibration method. Positioning errors are reduced to less than one millimeter.

CHAPTER 4 IBVS WITH LASER SYSTEM

4.1 Introduction

It is well known that traditional IBVS has some weakness, including image singularities, image local minima and relative long trajectory in Cartesian space. In this chapter, a new approach to switching control of IBVS with laser system is introduced to conquer those disadvantages.

The objective of the research is using an eye in hand system with laser pointer to control the position of the end effector with respected to the static object as required by the task. The simple off the shelf laser pointer based triangulation method is applied to realize the depth estimation. Furthermore, the laser spot can be considered as an image feature easing the image processing. The switching rules are introduced in detail.

4.2 Traditional IBVS

A review of the establishment of the image Jacobian matrix and the control law are briefly given in this section.

First of all, the velocity screw of the camera is defined as

$$\dot{r} = \begin{bmatrix} {}^{C}v_{C} & {}^{C}\omega_{C} \end{bmatrix}^{T} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^{T} \tag{4-1}$$

The image feature is represented as

$$f = \begin{bmatrix} u & v \end{bmatrix}^{T} \tag{4-2}$$

and

$$\dot{f} = \begin{bmatrix} \dot{u} & \dot{v} \end{bmatrix}^{T} \tag{4-3}$$

represents the corresponding image feature velocity.

The depth of each feature point is defined as

$$Z = [Z_1 \quad \cdots \quad Z_n]^T \tag{4-4}$$

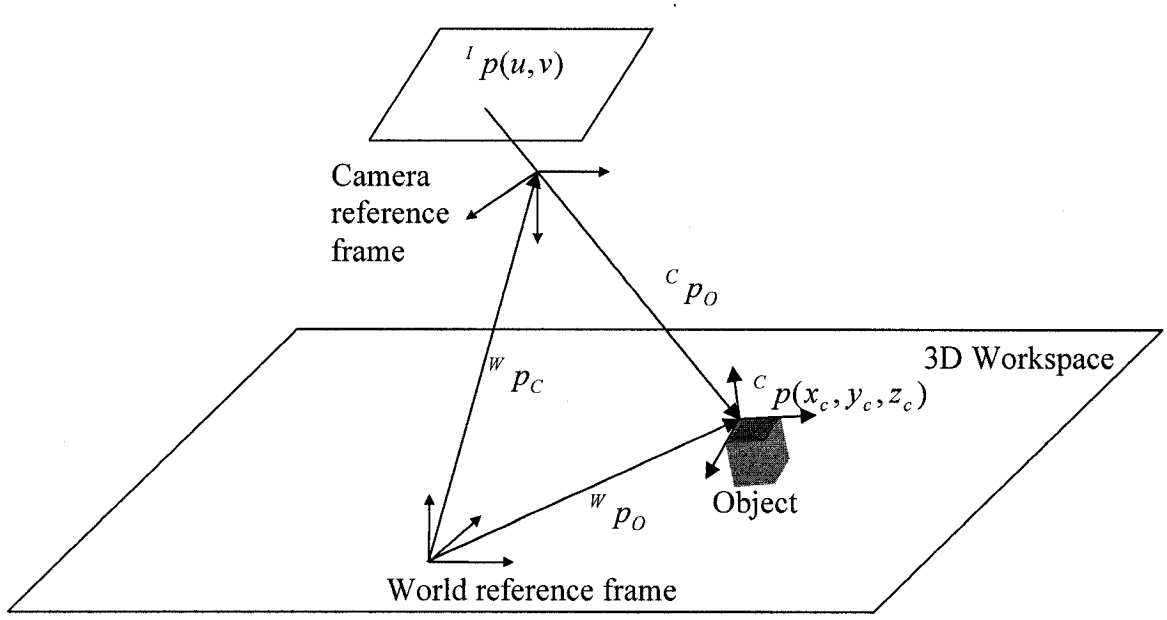


Figure 4-1 Reference frames and object coordinate

In order to create the feedback control for robot based on the velocity of feature points, the relationship between motion of the image features and the motion of the camera is given by

$$\dot{f} = \mathbf{J}_{img}(f, Z)\dot{r} \tag{4-5}$$

where

$$\mathbf{J}_{img}(f, Z) = \begin{bmatrix} \mathbf{J}_{img}(f_1, Z_1) \\ \vdots \\ \mathbf{J}_{img}(f_n, Z_n) \end{bmatrix} \tag{4-6}$$

is the image Jacobian matrix. To derive this matrix first we define a point

$$^C p_O = [x_c \quad y_c \quad z_c]^T \tag{4-7}$$

where (x_c, y_c, z_c) is the camera coordinates. The time derivative of ${}^{C}p_O$ is given as

$$
\begin{aligned}
{}^{C}\dot{p}_O &= \frac{d^{C}p_O}{dt} \\
&= \frac{d}{dt}\left\{{}^{C}\mathbf{R}_W({}^{W}p_O - {}^{W}p_C)\right\} \\
&= {}^{C}\dot{\mathbf{R}}_W({}^{W}p_O - {}^{W}p_C) + {}^{C}\mathbf{R}_W({}^{W}\dot{p}_O - {}^{W}\dot{p}_C)
\end{aligned} \tag{4-8}
$$

where ${}^{C}\mathbf{R}_W$ is the rotation matrix from camera reference frame to world reference frame, and as shown in Figure 4-1, ${}^{W}p_O$ is the relative distance of the feature point with respect to world reference frame and ${}^{W}p_C$ is the location of the camera reference frame written in world reference frame. For a stationary target, ${}^{W}\dot{p}_O$ is set to zero.

Considering the property of the skew-symmetric matrix

$$
{}^{C}\dot{\mathbf{R}}_W = {}^{C}\mathbf{R}_W\mathbf{sk}^{T}({}^{W}\omega_C), \tag{4-9}
$$

one can deduce ${}^{C}\dot{p}_O$ as follows

$$
\begin{aligned}
{}^{C}\dot{p}_O &= {}^{C}\mathbf{R}_W\mathbf{sk}^{T}({}^{W}\omega_C)({}^{W}p_O - {}^{W}p_C) - {}^{C}\mathbf{R}_W{}^{W}\dot{p}_C \\
&= {}^{C}\mathbf{R}_W\left\{-\mathbf{sk}({}^{W}\omega_C)({}^{W}p_O - {}^{W}p_C)\right\} - {}^{C}\mathbf{R}_W{}^{W}v_C \\
&= {}^{C}\mathbf{R}_W\left\{-{}^{W}\omega_C \times ({}^{W}p_O - {}^{W}p_C)\right\} - {}^{C}v_C \\
&= -{}^{C}\mathbf{R}_W{}^{W}\omega_C \times \left\{{}^{C}\mathbf{R}_W({}^{W}p_O - {}^{W}p_C)\right\} - {}^{C}v_C \\
&= -{}^{C}\omega_C \times {}^{C}p_O - {}^{C}v_C \\
&= \mathbf{sk}({}^{C}p_O){}^{C}\omega_C \times {}^{C}p_O - {}^{C}v_C \\
&= \begin{bmatrix} -\mathbf{I} & \mathbf{sk}({}^{C}p_O) \end{bmatrix} \begin{bmatrix} {}^{C}v_C \\ {}^{C}\omega_C \end{bmatrix}
\end{aligned}
$$

$$= \begin{bmatrix} -1 & 0 & 0 & 0 & -z_c & y_c \\ 0 & -1 & 0 & z_c & 0 & -x_c \\ 0 & 0 & -1 & -y_c & x_c & 0 \end{bmatrix} \begin{bmatrix} {}^C v_C \\ {}^C \omega_C \end{bmatrix} \tag{4-10}$$

When neglecting the camera distortion, one has the transformation between pixel indices (u, v) and the coordinates with respect to the camera reference frame (x_c, y_c, z_c) presented in (2-8). The time derivative of (2-8) is given as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{s_x} & 0 \\ 0 & -\frac{\lambda}{s_y} \end{bmatrix} \begin{bmatrix} \frac{1}{z_c} & 0 & -\frac{x_c}{z_c^2} \\ 0 & \frac{1}{z_c} & -\frac{y_c}{z_c^2} \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{z}_c \end{bmatrix} \tag{4-11}$$

Substituting (4-10) into (4-11) obtains

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{s_x} & 0 \\ 0 & -\frac{\lambda}{s_y} \end{bmatrix} \begin{bmatrix} \frac{1}{z_c} & 0 & -\frac{x_c}{z_c^2} \\ 0 & \frac{1}{z_c} & -\frac{y_c}{z_c^2} \end{bmatrix} {}^C \dot{p}_O$$

$$= \begin{bmatrix} -\frac{\lambda}{s_x} & 0 \\ 0 & -\frac{\lambda}{s_y} \end{bmatrix} \begin{bmatrix} \frac{1}{z_c} & 0 & -\frac{x_c}{z_c^2} \\ 0 & \frac{1}{z_c} & -\frac{y_c}{z_c^2} \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 & -z_c & y_c \\ 0 & -1 & 0 & z_c & 0 & -x_c \\ 0 & 0 & -1 & -y_c & x_c & 0 \end{bmatrix} \begin{bmatrix} {}^C v_C \\ {}^C \omega_C \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{s_x} & 0 \\ 0 & \frac{1}{s_y} \end{bmatrix} \begin{bmatrix} \frac{\lambda}{z_c} & 0 & -\lambda\frac{x_c}{z_c^2} & -\lambda\frac{x_c y_c}{z_c^2} & \lambda + \lambda\frac{x_c^2}{z_c^2} & -\lambda\frac{y_c}{z_c} \\ 0 & \frac{\lambda}{z_c} & -\lambda\frac{y_c}{z_c^2} & -\lambda - \lambda\frac{x_c^2}{z_c^2} & \lambda\frac{x_c y_c}{z_c^2} & \lambda\frac{x_c}{z_c} \end{bmatrix} \begin{bmatrix} {}^C v_C \\ {}^C \omega_C \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{s_x} & 0 \\ 0 & \frac{1}{s_y} \end{bmatrix} \begin{bmatrix} \frac{\lambda}{z_c} & 0 & -\frac{x_i}{z_c} & -\frac{x_i y_i}{\lambda} & \frac{\lambda^2 + x_i^2}{\lambda} & -y_i \\ 0 & \frac{\lambda}{z_c} & -\frac{y_i}{z_c} & \frac{-\lambda^2 - x_i^2}{\lambda} & \frac{x_i y_i}{\lambda} & x_i \end{bmatrix} \begin{bmatrix} {}^C v_C \\ {}^C \omega_C \end{bmatrix} \tag{4-12}$$

where $[x_i \quad y_i]^T$ are the image coordinates in camera reference frame. Considering the effective size of the pixel (s_x, s_y) are constant coefficients, we set them equal to one to make the equations clear. The transformation between $[x_i \quad y_i]^T$ and the pixel indices $[u \quad v]^T$ has been presented in (2-6) and (2-7). Rewriting them to matrix form gets

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} -1 & 0 & u_0 \\ 0 & -1 & v_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4-13}$$

For one feature point, the velocity of it seen from the camera reference frame is presented as follows

$$\mathbf{J}_{img}(f_i, Z_i) = \begin{bmatrix} \frac{\lambda}{Z_i} & 0 & -\frac{x_i}{Z_i} & -\frac{x_i y_i}{\lambda} & \frac{\lambda^2 + x_i^2}{\lambda} & -y_i \\ 0 & \frac{\lambda}{Z_i} & -\frac{y_i}{Z_i} & \frac{-\lambda^2 - y_i^2}{\lambda} & \frac{x_i y_i}{\lambda} & x_i \end{bmatrix} \tag{4-14}$$

The control law can be established by considering the simplest proportional control as

$$e = K\mathbf{J}_{img}^{+}(f, Z)\dot{f} \tag{4-15}$$

where K is the proportional gain, $\mathbf{J}_{img}^{+}(f, Z)$ is the pseudo inverse of the image Jacobian and $e = \dot{r}$ is the vision based task which is being regulated to zero.

Since generally the input of the robot controller is the joint angle, the relationship between camera velocity and the joint velocity needs to be derived. First of all, the camera velocity can be transferred to end effector velocity as follows,

$$\begin{bmatrix} {}^E v_E \\ {}^E \omega_E \end{bmatrix} = \begin{bmatrix} {}^E\mathbf{R}_C & -{}^E\mathbf{R}_C \mathbf{sk}({}^C T_E) \\ 0 & {}^E\mathbf{R}_C \end{bmatrix} \begin{bmatrix} {}^C v_C \\ {}^C \omega_C \end{bmatrix} \tag{4-16}$$

where $\begin{bmatrix} {}^{E}v_{E} & {}^{E}\omega_{E} \end{bmatrix}^{T}$ is the velocity screw of end effector in end effector frame, ${}^{E}\mathbf{R}_{C}$ is the rotation matrix from end effector frame to camera frame, and ${}^{C}T_{E}$ is the translational vector from camera frame to end effector frame.

The joint velocity is related to the end effector velocity by robot Jacobian matrix represented as:

$$\dot{q} = \mathbf{J}_{r}^{-1} \begin{bmatrix} {}^{E}v_{E} \\ {}^{E}\omega_{E} \end{bmatrix} \tag{4-17}$$

where $\mathbf{J}_{r}^{-1}$ is the inverse robot Jacobian matrix in end effector frame. According to (4-16) and (4-17), the camera velocity in camera frame is transferred to robot joint velocity.

4.3 Depth Estimation

Refer to (4-14), $\mathbf{J}_{img}(f, Z)$ is a function of features and their depth. A number of depth estimation methods have been proposed for IBVS. One of the solutions is choosing the desired image features and their depth for a constant image Jacobian, however it is proved stable only in a neighborhood of the desired position. Another solution is on line estimate depth of every feature point [39]. In this research, considering one of the methods to measure the distance from camera to targets is through the use of laser triangulation, we solve the depth estimation by a simple off the shelf laser pointer. Moreover the laser spot can be chosen as an image feature. It is highly easy for image processing by using a fixed-level thresholding.

Assume that the laser beam is lying in the same plane with the camera optical axis. In this configuration, no matter how camera is moving, the trajectory of the laser spot in the image is a straight line passing the principal point in image. If we consider the laser

pointer as a camera and the laser beam is its optical axis, the straight line in the image will be the epipolar line of the imaginary stereo configuration.

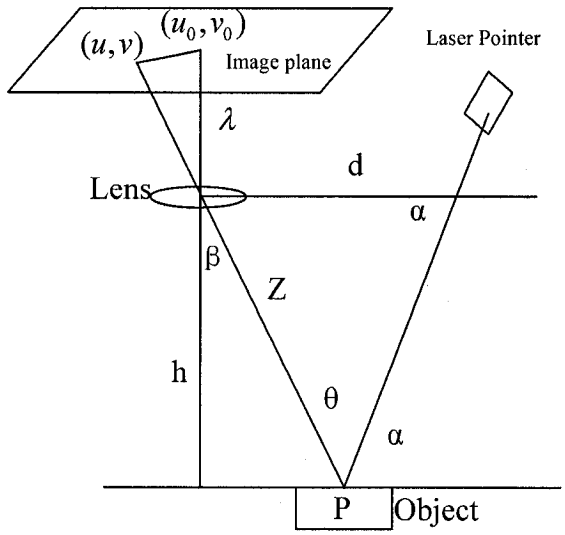


Figure 4-2 Calculate the Depth of a Point by Using Triangulation

As shown in Figure 4-2, d denotes the horizontal distance between the laser beam and the optical axis of lens of the camera, and α is the angle the laser beam makes with the horizontal. Both of them are fixed when the ranging system is installed. P is the intersecting point of the laser beam and the object surface.

Apply the trigonometry to derive a function of depth with respect to pixel indices (u, v). The triangle inside the camera gives

$$\beta = \pm a\tan\left(\frac{\sqrt{(u-u_0)^2 s_x^2 + (v-v_0)^2 s_y^2}}{\lambda}\right) \tag{4-18}$$

where (u, v) is the pixel index of a point, (u_0, v_0) is the principal point, and (s_x, s_y) is the effective size of the pixel.

Applying the Law of Sines to the triangle (lens-P-laser) yields

$$\frac{Z}{\sin\alpha} = \frac{d}{\sin\theta} \tag{4-19}$$

in which θ is the inclination of incident ray and reflected ray, and Z is the distance between P and the camera lens. θ is obtained as:

$$\theta = 180° - \alpha - (90° - \beta) = 90° + \beta - \alpha \tag{4-20}$$

Substituting (4-20) into (4-19) gives

$$Z = \frac{d \sin \alpha}{\sin \theta} = \frac{d \sin \alpha}{\cos(\alpha - \beta)} \tag{4-21}$$

Here the Z can be used to approximate the depth of each feature point in the object surface.

4.4 Image Based Visual Servoing with Laser Pointer

It is well known that, for IBVS, since the control law is only in the image plane, unnecessary motions of end effector are performed. Furthermore there may have image singularities and image local minima which may lead to potential failure. In this section, a new approach to IBVS with laser pointer is presented in detail to conquer the above drawbacks.

The proposed algorism is composed of three sub-controllers, which are driving the laser spot on the object, combining IBVS with laser spot for servoing, and applying a constant image Jacobian at the desired camera configuration to reach the desired position. The system configuration is shown in Figure 4-3, which is composed of 6DOF robot, a camera mounted on the robot end effector and a laser pointer rigidly linked to the camera. The object is assumed to be static with respect to robot reference frame.

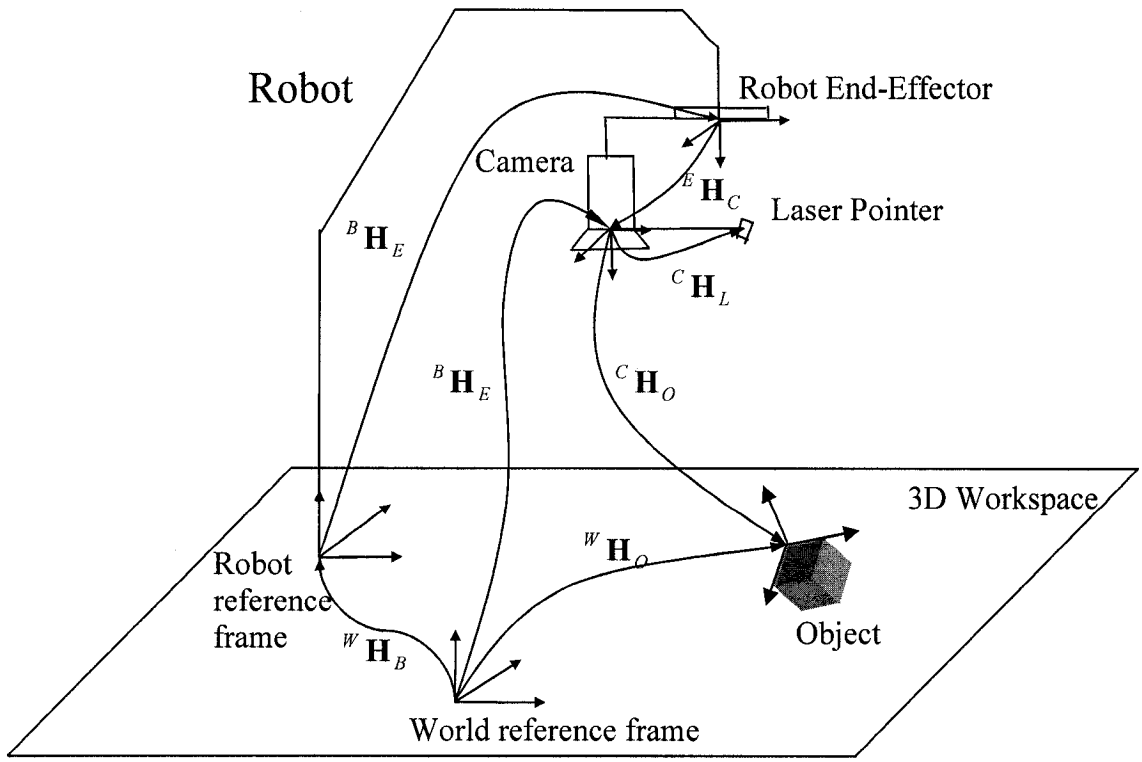


Figure 4-3 IBVS with laser system configuration

The block diagram of IBVS with laser pointer control system is presented in Figure 4-4.

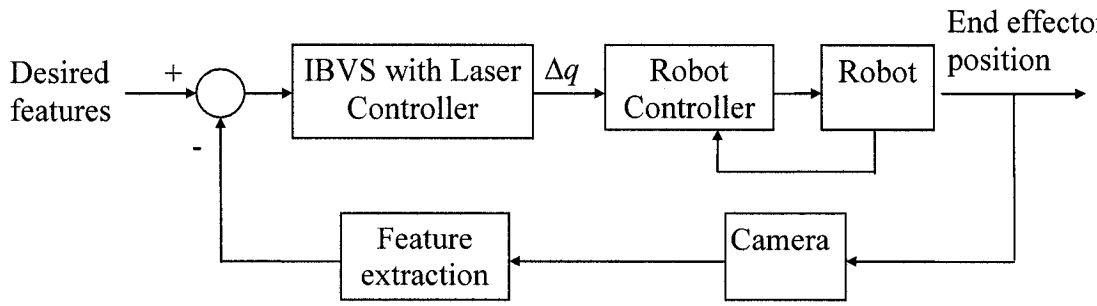


Figure 4-4 Block diagram of IBVS with laser control system

Stage 1: Drive the Laser Spot on the Object

To project the laser spot on the object, there are two kinds of situations need to be considered. One is that all features are in the field of view (Stage 1.A), and the other is

that partial object is in the field of view (Stage 1.B). Both of them will be discussed below in detail. In Stage 1.C, the velocity transformed from camera frame to end effector frame is derived.

Stage 1.A: All Features in the Field of View

Equation (4-5) can be decomposed into translational and rotational component parts as shown below

$$\dot{f} = \left[\mathbf{J}^{t}_{img}(f,Z) \quad \mathbf{J}^{r}_{img}(f)\right] \begin{bmatrix} {}^{C}v_{C} \\ {}^{C}\omega_{C} \end{bmatrix} \tag{4-22}$$

where $\mathbf{J}^{t}_{img}(f,Z)$ and $\mathbf{J}^{r}_{img}(f)$ are stacked by $\mathbf{J}^{t}_{img}(f_i,Z_i)$ and $\mathbf{J}^{r}_{img}(f_i)$ given by

$$\mathbf{J}^{t}_{img}(f_i,Z_i) = \begin{bmatrix} \frac{\lambda}{Z_i} & 0 & -\frac{x_i}{Z_i} \\ 0 & \frac{\lambda}{Z_i} & -\frac{y_i}{Z_i} \end{bmatrix} \tag{4-23}$$

$$\mathbf{J}^{r}_{img}(f_i) = \begin{bmatrix} -\frac{x_i y_i}{\lambda} & \frac{\lambda^2 + x_i^{\,2}}{\lambda} & -y_i \\ \frac{-\lambda^2 - y_i^{\,2}}{\lambda} & \frac{x_i y_i}{\lambda} & x_i \end{bmatrix} \tag{4-24}$$

It is noted that $\mathbf{J}^{t}_{img}$ is related to both the features and their depth, and $\mathbf{J}^{r}_{img}$ is only a function of the image features. Since the laser pointer is mounted on the robot end effector, it is possible to control it by performing 3DOF rotational motion. It is known that 2DOF of rotational motion is enough for a laser pointer to project its dot image on the desired target. Here the reason of using 3DOF instead of 2DOF is that the rotation along the camera Z axis can be used to solve the rotation problem such as 180 deg around the optical axis, which is well known causing the image singularity presented by F.Chaumette [16]. Since each feature point expects to move to the symmetrical point with

respect to the image center with shortest path in image, it will lead to a pure backward translational camera motion along the optical axis to infinity. Consequently, equation (4-14) becomes

$$\mathbf{J}_{img}(f_i, Z_i) = \begin{bmatrix} 0 & 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & -\lambda & 0 & 0 \end{bmatrix} \tag{4-25}$$

which is rank 2 and will cause image singularity. To avoid this particular condition, imaginary desired features are designed. According to the desired image including object features and the desired laser spot, a new image can be designed, which is shown in Figure 4-5. Let the distance between desired and current laser spot be d_l, then the imaginary object features are obtained by shifting all desired object features d_l to the current laser spot.

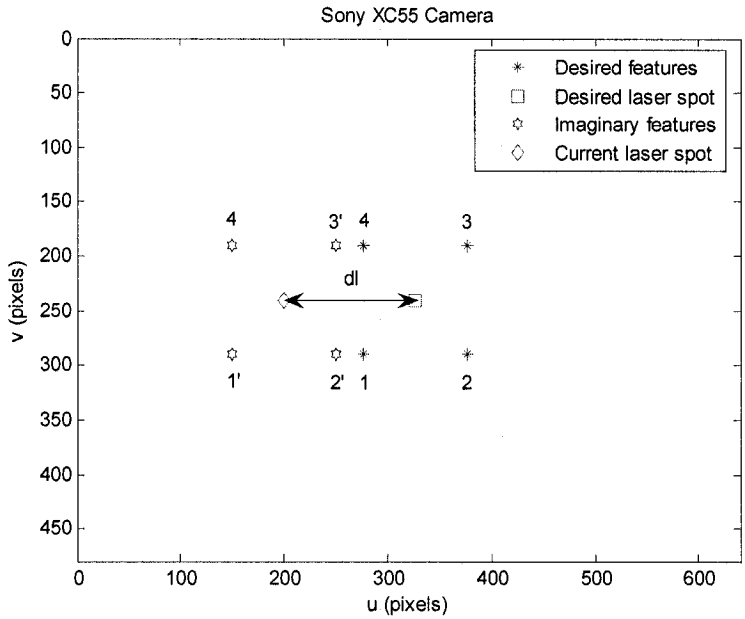


Figure 4-5 Example of creating imaginary features

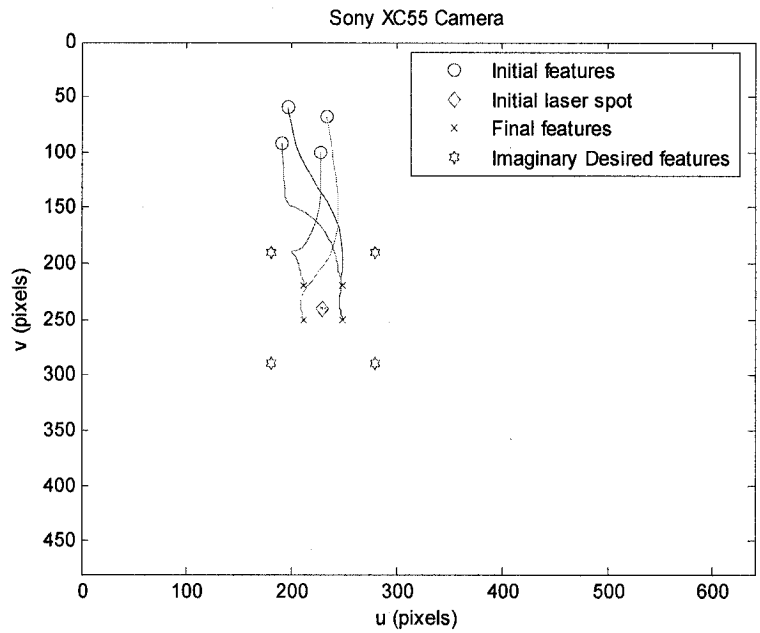


Figure 4-6 Image trajectory of driving the laser spot on the object

It is assumed that the height of the object is relatively small. Therefore, when the laser spot moves from the platform to the surface, there is no big discrepancy of the laser spot in the image.

The control goal is to minimize the summation error between final features and the imaginary features, as presented below

$$\min\left\{\sum(f_{id} - f)\right\} \tag{4-26}$$

where f_{id} is the imaginary desired features.

The switching rule is described as: when the error norm falls below a predetermined threshold value, the controller will switch from first stage to second stage. The switching condition is given by

$$\left\|f_{id} - f\right\| \leq f_0 \tag{4-27}$$

where f_0 is the predetermined threshold value. Figure 4-6 presents a desired image of this stage as an example.

Therefore the relationship between the motion of the image feature and the rotation DOF of camera is presented as

$$\dot{f} = \begin{bmatrix} 0 & \mathbf{J}^r_{img}(f) \end{bmatrix} \begin{bmatrix} {}^C v_C \\ {}^C \omega_C \end{bmatrix} \tag{4-28}$$

The proportional control law is given by

$$\begin{aligned} e &= K_l \begin{bmatrix} 0 & \mathbf{J}^r_{img}(f) \end{bmatrix}^+ \dot{f} \\ &= K_l \begin{bmatrix} 0 & \mathbf{J}^r_{img}(f) \end{bmatrix}^+ [f_{id} - f] \end{aligned} \tag{4-29}$$

where K_l is the proportional gain.

Notice that the function of this control law not only drives the laser spot on the object but also solves the image singularity problem. As mentioned in [16], a pure rotation of 180 deg around the optical axis will lead to image singularity and will cause a pure backward translational camera motion along the optical axis. In the proposed algorism, the rotation of the camera is mainly executed in the first stage of control, which is 3DOF rotational motion. The second stage of control is primarily a translation. Hence, the backward translational camera motion will be avoided.

Stage 1.B: Partial Object in the Field of View

When only partial object is in the field of view, some features are not available. Therefore, the idea is to control 2DOF rotational motion of camera to project the laser spot on the centroid of the object in image plane until all the features are in the field of

view. The area centroid is the centroid of the area occupied by pixels satisfying the required threshold. According to this idea, Equation (4-5) can be rewritten as:

$$\dot{f} = \mathbf{J}^{r}_{img}(f)\begin{bmatrix}\omega_x \\ \omega_y\end{bmatrix} \tag{4-30}$$

where $\mathbf{J}^{r}_{img}(f)$ is stacked by $\mathbf{J}^{r}_{img}(f_i)$ given by

$$\mathbf{J}^{r}_{img}(f_i) = \begin{bmatrix} -\frac{x_i y_i}{\lambda} & \frac{\lambda^2 + x_i^{\,2}}{\lambda} \\ \frac{-\lambda^2 - y_i^{\,2}}{\lambda} & \frac{x_i y_i}{\lambda} \end{bmatrix} \tag{4-31}$$

As mentioned before, 2DOF of rotational motion is enough for a laser pointer to project its dot image on the desired target. Thus the other elements of the camera velocity screw are set to zero.

The proportional control law is given by

$$e = K_{l2}\mathbf{J}^{r+}_{img}(f)\dot{f} \tag{4-32}$$

where K_{l2} is the proportional gain, $\mathbf{J}^{r+}_{img}(f)$ is the pseudo inverse of $\mathbf{J}^{r}_{img}(f)$.

To attain the area centroid of the partial object in image, generally we calculate the first order moments. Let R present the region of the partial object in a binary image $I(k, j)$, which can be obtain by fixed-level thresholding. For a digital image, the moments of the region R are defined as:

$$m_{kj} = \sum_{(x,y)\in R} x^k y^j \tag{4-33}$$

$$k \geq 0, \quad j \geq 0$$

where (x, y) represent the row and column of a pixel in the region R respectively. According to the moments definition, we have the area of the region R and the centroid of R as:

$$A_R = m_{00} \tag{4-34}$$

$$f_c = \begin{bmatrix} f_{xc} \\ f_{yc} \end{bmatrix} = \begin{bmatrix} \frac{m_{10}}{m_{00}} \\ \frac{m_{10}}{m_{00}} \end{bmatrix} \tag{4-35}$$

where f_c is the centroid of the partial object in image and the desired feature as well. Thus we have the proportional control law shown as following:

$$e = K_{l2} \mathbf{J}_{img}^{r+}(f) \ [f_c - f] \tag{4-36}$$

When all the feature points are in the field of view, the control law will switch to (4-29).

Stage 1.C: Transform Velocity from Camera Frame to End Effector Frame

In the real case, performing 3DOF rotational motion of the camera will cause the robot end effector unnecessary movement. To solve this problem, consider that the camera velocity can be transferred to end effector velocity as shown in (4-16).

It is assumed that the optical axis of the camera is coincident with the Z axis of the end effector. In order to perform only 3DOF rotational motion of the end effector, the ${}^E v_E$ needs to be set to zero. According to (4-16) the linear velocity screw of camera in camera frame can be calculated as follows:

$$\begin{aligned} {}^E v_E &= {}^E\mathbf{R}_C \, {}^C v_C - {}^E\mathbf{R}_C \mathbf{sk}({}^C T_E) \, {}^C \omega_C \\ &= {}^E\mathbf{R}_C \, {}^C v_C - {}^E\mathbf{R}_C \, {}^C T_E \times {}^C \omega_C = 0 \end{aligned} \tag{4-37}$$

Therefore,

$$^{C}v_{C} = {}^{C}T_{E} \times {}^{C}\omega_{C} \tag{4-38}$$

where $^{C}\omega_{C}$ has already been calculated in (4-29) or (4-36).

Stage 2: Using Object Features and Laser Spot for Image Jacobian

The key problem for Stage 2 of IBVS with laser pointer algorism is how to obtain the image Jacobian to relate the motion of the laser image feature and the motion of the laser spot. According to the derivation of traditional image Jacobian matrix, the target is supposed to be stable. In our system, since the laser pointer moves together with the camera, normally the laser spot is not stationary. However, during the motion of the camera, the laser spot on the target surface can be focused on the approximately same point by controlling three rotational DOF of the robot end effector. Hence the laser spot on the target can be considered as a stationary point adapting to image Jacobian matrix of traditional IBVS. Based on above scheme, the algorithm is presented in detail as follows.

Let

$$f = \begin{bmatrix} f_1 & \cdots & f_n & f^l & f^l \end{bmatrix}^T \tag{4-39}$$

represent the image feature including the laser spot $f^l = \begin{bmatrix} x^l & y^l \end{bmatrix}^T$. The modified relationship between the motion of the image feature and the motion of camera is given by

$$\dot{f} = \mathbf{J}_{img}(f, Z)\dot{r} \tag{4-40}$$

where $\mathbf{J}_{img}(f, Z)$ can also be decomposed into translational and rotational component parts as shown below

$$\mathbf{J}_{img}(f,Z)=\begin{bmatrix}\mathbf{J}^{t}_{img}(f_i,f^{l},Z) & 0\\ 0 & \mathbf{J}^{r}_{img}(f^{l})\end{bmatrix} \tag{4-41}$$

In equation (4-41),

$$\mathbf{J}^{t}_{img}(f_i,f^{l},Z)=\begin{bmatrix}\mathbf{J}^{t}_{img}(f_1,Z)\\ \vdots\\ \mathbf{J}^{t}_{img}(f_n,Z)\\ \mathbf{J}^{t}_{img}(f^{l},Z)\end{bmatrix} \tag{4-42}$$

in which $\mathbf{J}^{t}_{img}(f_i,Z)$ and $\mathbf{J}^{t}_{img}(f^{l},Z)$ are presented in (4-23). The image Jacobian matrix $\mathbf{J}^{r}_{img}(f^{l})$ in (4-41) is calculated by (4-24).

The switching rule is described as: if the error norm of current laser spot and desired laser spot falls below a threshold, the IBVS with laser controller will switch from second stage to third stage, which is given by

$$\left\|f^{l}_{d}-f^{l}\right\|\le f^{l}_{0} \tag{4-43}$$

where f^{l}_{d} is the desired laser spot feature and f^{l}_{0} is the predetermined threshold value. This threshold is directly related to the depth that will affect the stability of the controller. Thus the selection of this threshold needs to be relatively small.

The proportional control law is given by

$$e=K\mathbf{J}^{+}_{img}(f,Z)\dot{f} \tag{4-44}$$

where K is the proportional gain, $\mathbf{J}^{+}_{img}(f,Z)$ is the pseudo inverse of $\mathbf{J}_{img}(f,Z)$.

Referring to (4-41), $\dot{f}$ is the desired feature motion in the image, which can be defined as an error function

$$e(f)=f_d-f \tag{4-45}$$

where

$$f_d = \begin{bmatrix} f_{1d} & \cdots & f_{nd} & f_d^l & f(f_1,\cdots,f_n) \end{bmatrix}^T \tag{4-46}$$

is the desired image feature, in which the function of the last element can be defined as

$$f(f_1,\cdots,f_n) = ((x_1 + \cdots + x_n)/n, (y_1 + \cdots + y_n)/n) \tag{4-47}$$

The function also can be defined as the center of gravity of the planar surface in the image plane. In (4-47), $\begin{bmatrix} x_n & y_n \end{bmatrix}^T$ is the current image feature.

Thus equation (4-44) can be rewritten as follows

$$e = K \left(\begin{bmatrix} \mathbf{J}_{img}^t(f_1, Z) & 0 \\ \vdots & \vdots \\ \mathbf{J}_{img}^t(f_n, Z) & 0 \\ \mathbf{J}_{img}^t(f^l, Z) & 0 \\ 0 & \mathbf{J}_{img}^r(f^l) \end{bmatrix} \right)^+ \begin{bmatrix} f_{1d} - f_1 \\ \vdots \\ f_{nd} - f_n \\ f_d^l - f^l \\ f(f_1,\cdots,f_n) - f^l \end{bmatrix} \tag{4-48}$$

where the constant gain matrix K can be presented as

$$K = \begin{bmatrix} K_1 I & 0 \\ 0 & K_2 I \end{bmatrix} \tag{4-49}$$

in which K_1 and K_2 are the proportional gains for camera translational and rotational velocity respectively.

The example of an image trajectory in this stage is shown in Figure 4-7.

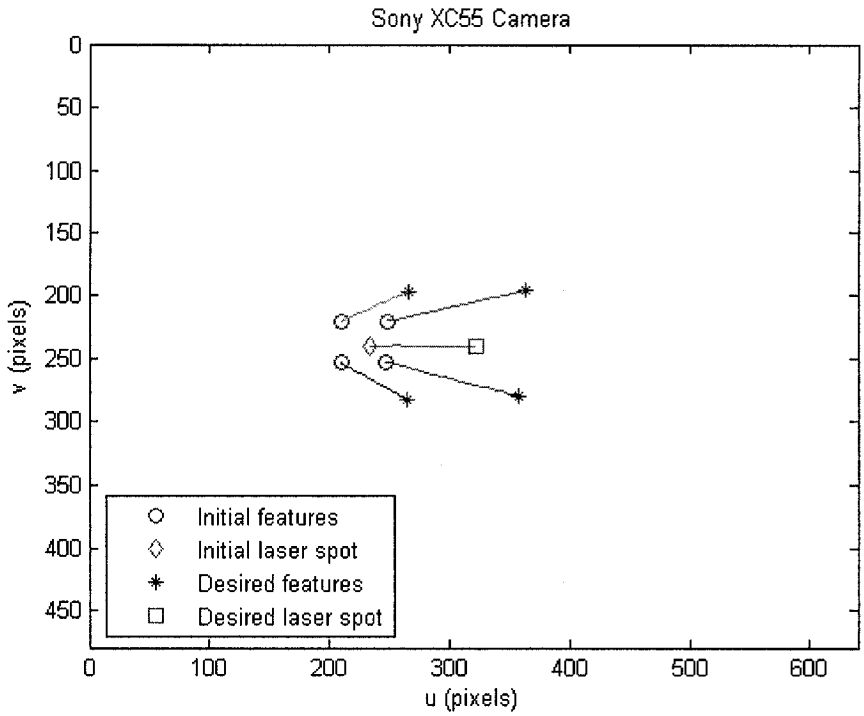


Figure 4-7 Image trajectory of the second control stage

Stage 3: Using Constant Image Jacobian

In this section, the third control stage is being introduced. In the proposed IBVS with laser algorithm, if the laser spot in the image plane is around its desired position, as presented in (4-43), the depths of the points will be approximate to the desired ones as well (essentially planary object). The IBVS with laser controller is switched to third stage, which uses traditional IBVS with constant image Jacobian.

$$\mathbf{J}_{img}(f_{id}, Z_{id}) = \begin{bmatrix} \frac{\lambda}{Z_{id}} & 0 & -\frac{x_{id}}{Z_{id}} & -\frac{x_{id}y_{id}}{\lambda} & \frac{\lambda^2 + x_{id}^{\ 2}}{\lambda} & -y_{id} \\ 0 & \frac{\lambda}{Z_{id}} & -\frac{y_{id}}{Z_{id}} & \frac{-\lambda^2 - y_{id}^{\ 2}}{\lambda} & \frac{x_{id}y_{id}}{\lambda} & x_{id} \end{bmatrix} \tag{4-50}$$

where f_{id} and Z_{id} are desired features and their corresponding depth respectively. The control law can be established as

$$e = K\mathbf{J}_{img}^{+}(f_d, Z_d)\dot{f} \tag{4-51}$$

The reason using constant image Jacobian instead of on-line updated image Jacobian is to avoid the image local minima. It has been presented that the constant image Jacobian at the desired camera configuration for IBVS can be used to reach the image global minimum [16]. So, as for our algorithm, when the laser spot is located at the desired position in the image, it is ensured that the current position is near a neighborhood of the desired position. In this situation, the laser spot will not be considered as image feature any more and the traditional IBVS with constant image Jacobian of desired image features will be applied.

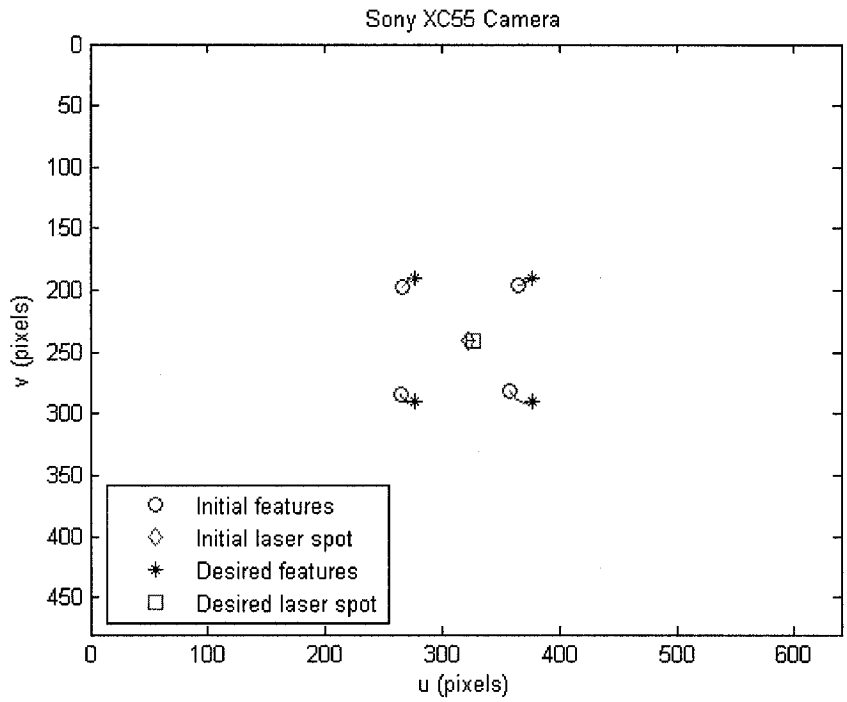


Figure 4-8 Image trajectory of the third control stage

The control goal of this stage is to reach the global minimum. The condition can be described as: if the feature error norm falls below a predetermined threshold, the whole IBVS with laser controller will be stopped. This threshold directly affects the accuracy of

the robot end effector pose with respect to object. The idea is choosing a relative bigger value to increase the convergent speed, when the accuracy is guaranteed. The example of an image trajectory in this stage is shown in Figure 4-8.

4.4 Summary

A new approach to switching control of IBVS with laser pointer is introduced in this chapter. A simple commercial laser pointer is used to calculate the depth of image feature. The control law and switching rules are introduced in detail.

The proposed system is derived from traditional IBVS. Therefore it inherits the advantages of IBVS, which are object model free and robust to the camera calibration and hand-eye calibration. In addition, by using laser pointer and partition DOF method, IBVS with laser system avoids the inherent drawbacks of traditional IBVS, image singularities and image local minima.

CHAPTER 5 SIMULATION AND EXPERIMENTAL RESULTS OF IBVS WITH LASER SYSTEM

5.1 Introduction

In this chapter, the proposed IBVS with laser system is simulated and experimentally tested. The first part is the simulation results of IBVS with laser pointer system. Four selected tasks are provided to cover most situations. The comparison with other methods and the robustness test are presented as well. In the second part, the experiment is through the robotic assembly to verify the effectiveness of IBVS with laser pointer system. A plastic object will be assembled in a metallic part. Both the ideal condition of the calibration and the robustness test are performed to fulfill the task.

5.2 Simulation Results of IBVS with Laser Pointer

The proposed algorithm has been simulated using Matlab and the Robotics Toolbox [69]. The simulation is based on the robot model of Motoman UPJ, the camera model of Sony XC55 and a cube object, all of which have been presented in Figure 4-3. The detailed camera parameters are shown in Table 3-1. The side length of the cube is 20mm and the four coplanar corners are chosen as the targets. The d and α for the fixed laser-camera configuration shown in Figure 4-2 are chosen as 30mm and 80deg respectively. The camera reference frame is set to be coincident with the robot end effector frame.

Four tasks are selected to evaluate the proposed method. The first task is, as presented in [1], a pure rotation π rad around the optical axis, which normally leads the traditional IBVS failure. The second and third tasks are relative small and large pose discrepancy

respectively. The initial image of the object in the last task is partially out of the field of view. The detailed parameters are shown in table below.

Table 5-1 Initial and desired poses of tasks

	Pose	X (m)	Y (m)	Z (m)	ϕ (rad)	θ (rad)	ψ (rad)
Task 1	Initial ${}^{C}\mathbf{H}_{O}$	0.011	-0.01	0.162	0	0	$-\pi$
	Desired ${}^{C}\mathbf{H}_{O}$	0.011	-0.01	0.162	π	0	$-\pi$
Task 2	Initial ${}^{C}\mathbf{H}_{O}$	-0.12	-0.05	0.42	-0.524	0.349	-2.793
	Desired ${}^{C}\mathbf{H}_{O}$	0.011	0.01	0.162	$-\pi/2$	0	$-\pi$
Task 3	Initial ${}^{C}\mathbf{H}_{O}$	-0.07	-0.08	0.44	0.175	-0.175	-2.444
	Desired ${}^{C}\mathbf{H}_{O}$	0.011	-0.01	0.162	π	0	$-\pi$
Task 4	Initial ${}^{C}\mathbf{H}_{O}$	-0.17	-0.05	0.42	-0.524	0.349	-2.793
	Desired ${}^{C}\mathbf{H}_{O}$	0.011	0.01	0.162	$-\pi/2$	0	$-\pi$

Table 5-2 Desired feature sets

Desired feature set 1	Desired object features (pixel)	$\begin{bmatrix} 277 & 290 \\ 377 & 290 \\ 377 & 190 \\ 327 & 240 \end{bmatrix}$
	Desired laser spot features (pixel)	$\begin{bmatrix} 327 & 240 \end{bmatrix}$
Desired feature set 2	Desired object features (pixel)	$\begin{bmatrix} 377 & 290 \\ 377 & 190 \\ 327 & 240 \\ 277 & 290 \end{bmatrix}$
	Desired laser spot features (pixel)	$\begin{bmatrix} 327 & 240 \end{bmatrix}$

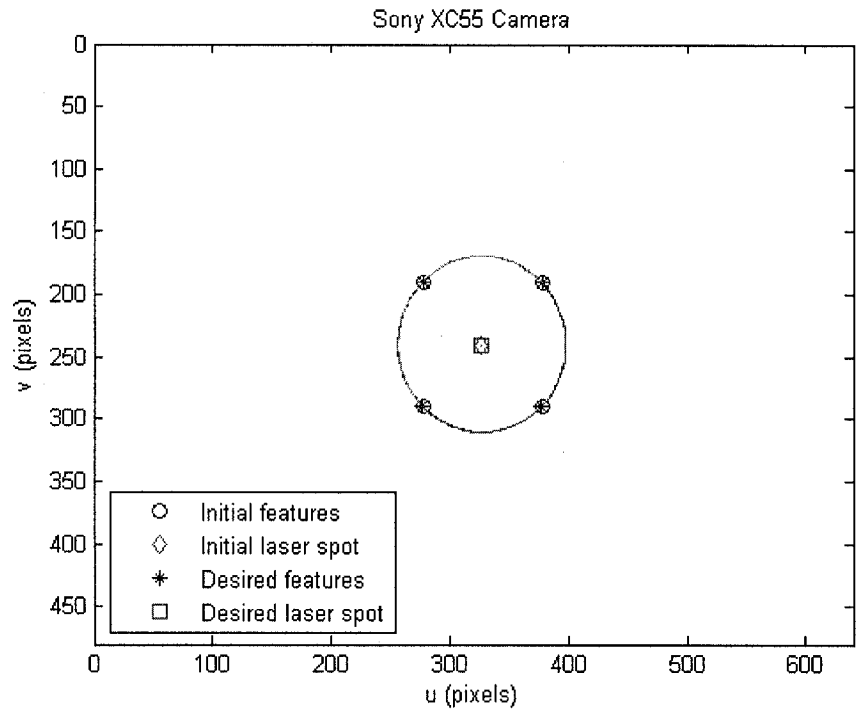
Two desired feature sets are created as shown in Table 5-2 to test two different rotation tasks. Task 1 and task 3 are involved in desired feature set 2, and task 2 and task

4 are involved in desired feature set 1. The predetermined threshold f_0 in (4-27) is set as 10 pixels and the condition of final feature error norm is set to 1 pixel. The f_0^l in (4-43) is tuned as 10 pixels, which is approximate to 10mm in the simulation environment. All the proportional gains of control law are set to 1.

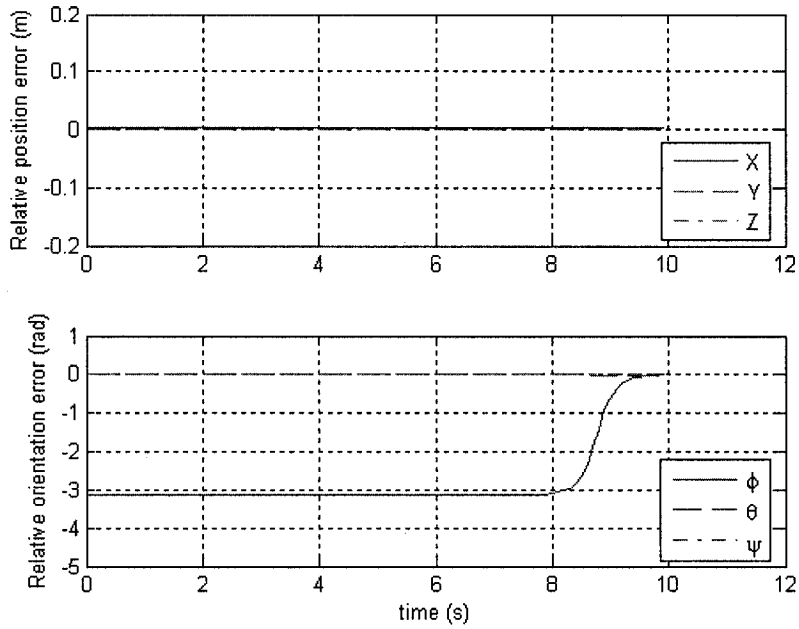
5.2.1 Results with Ideal Condition

Firstly, exact values of every transformation including camera model, hand-eye and robot world transform are known to test the IBVS with laser system.

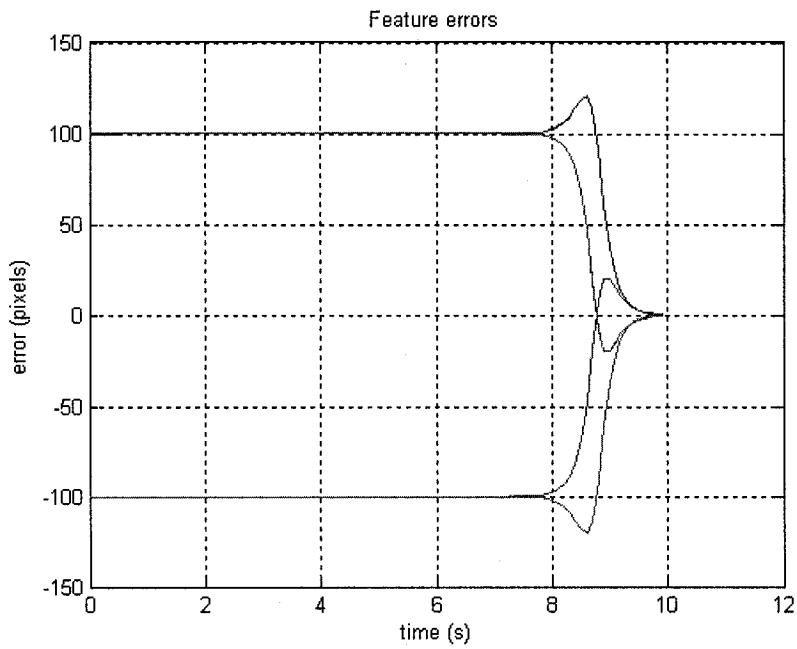
The simulation result of Task 1 that is a pure 180 deg rotational motion around the optical axis is shown in Figure 5-1.



(a) Image trajectory



(b) Relative position error and orientation error of robot end effector

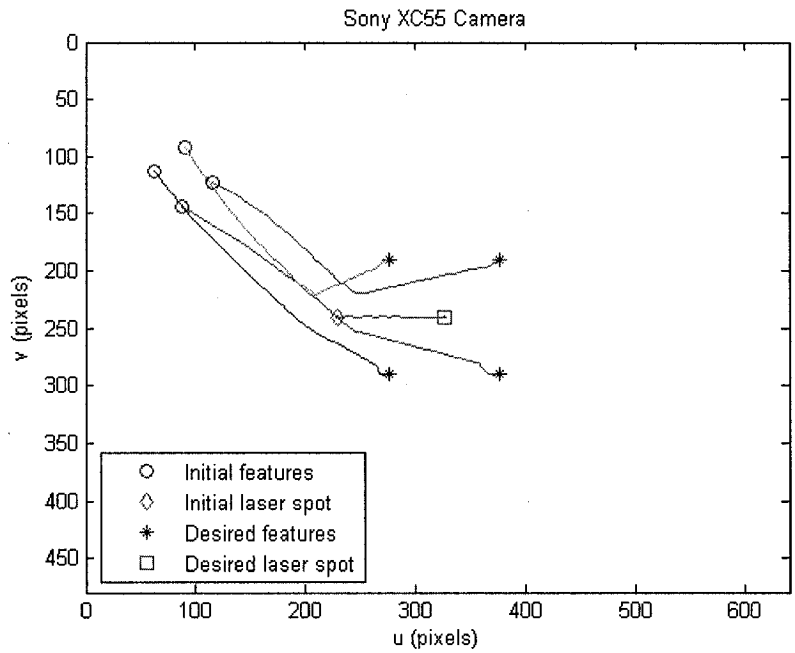


(c) Image feature errors

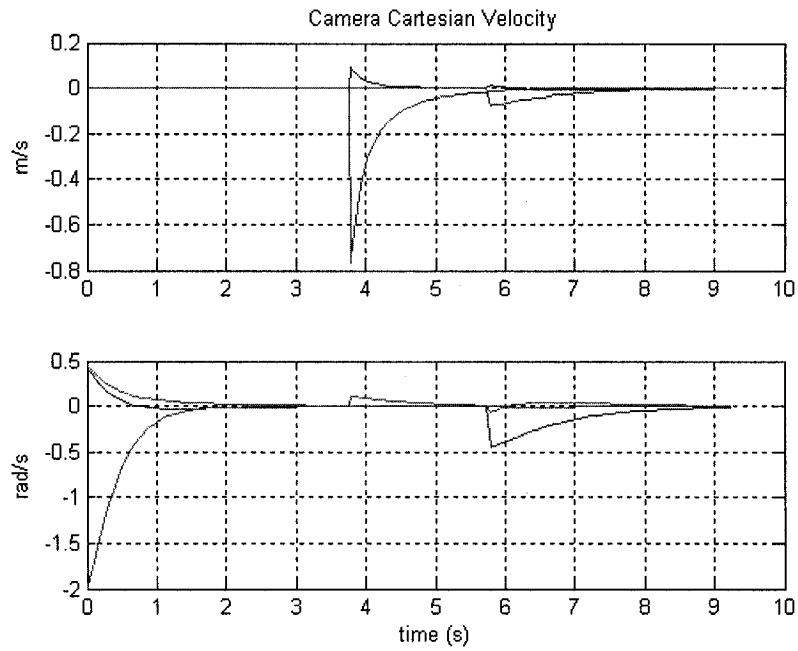
Figure 5-1 Simulation results of task 1

In Figure 5-1 (a), initial features have transformed to the symmetrical point with respect to the image center, and initial and desired positions in image are overlaid. It can be found that the relative orientation error in Figure 5-1 (b) is π rad. Instead of reducing the image feature errors as fast as possible by traditional IBVS, the image feature errors first increase a little then decrease (see Figure 5-1 c). Thus the image trajectory is a circle, and there is no translational motion in Cartesian space rather than a pure backward translational motion to infinity.

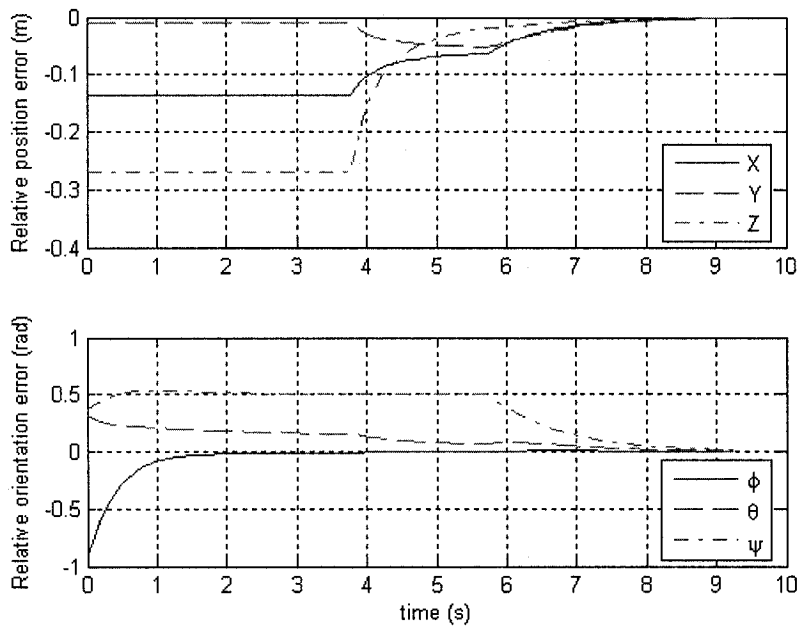
The second task is with a relative small pose discrepancy, which will be used to compare different methods in the comparison subsection. The relative position and orientation between initial position and desired position shown in Figure 5-2 (c) are within 0.3m and 1 rad respectively.



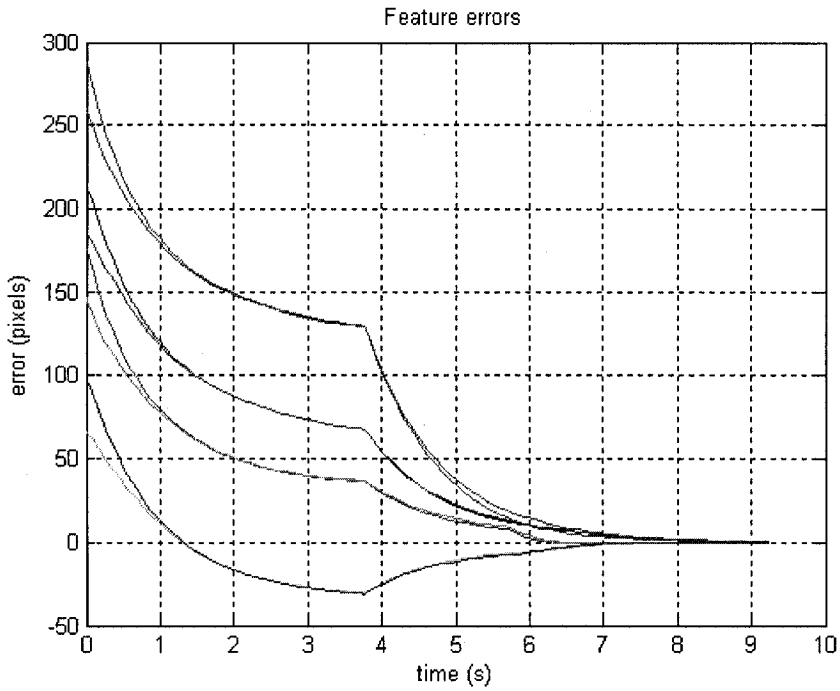
(a) Image trajectory



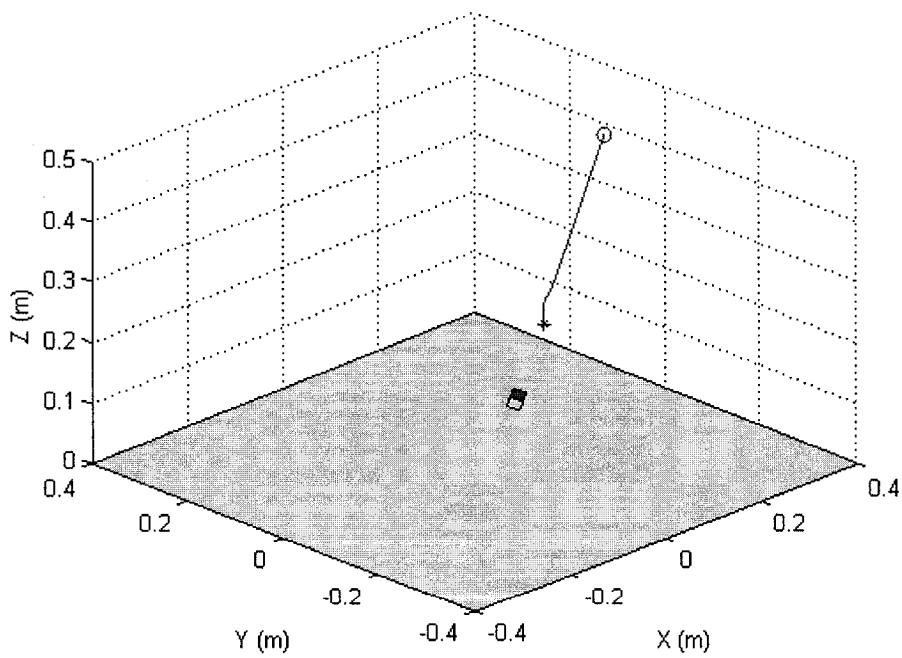
(b) Camera translational and rotational velocity in Cartesian space



(c) Relative position error and orientation error of robot end effector



(d) Image feature errors

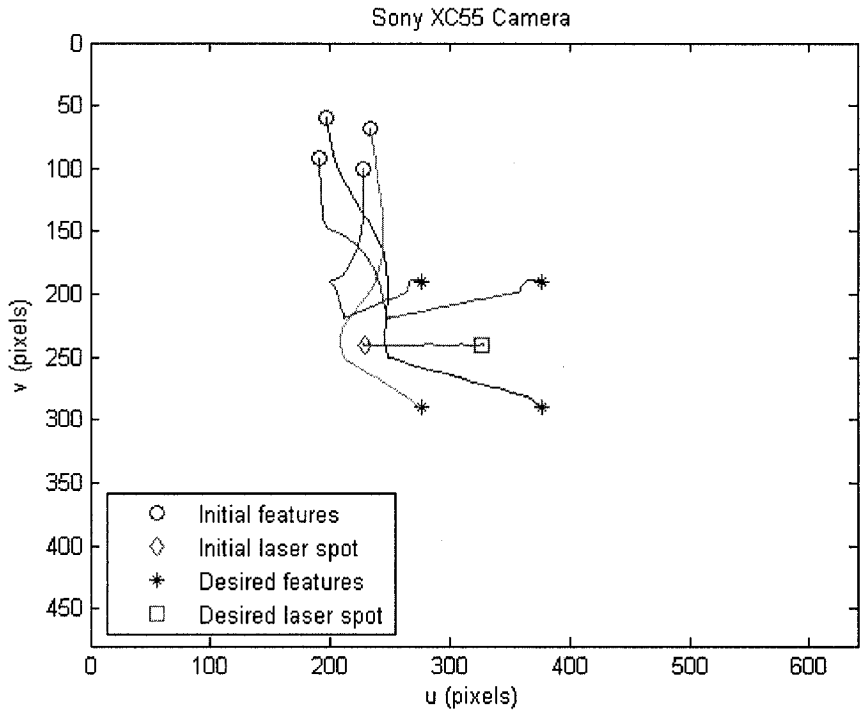


(e) End effector trajectory in Cartesian space

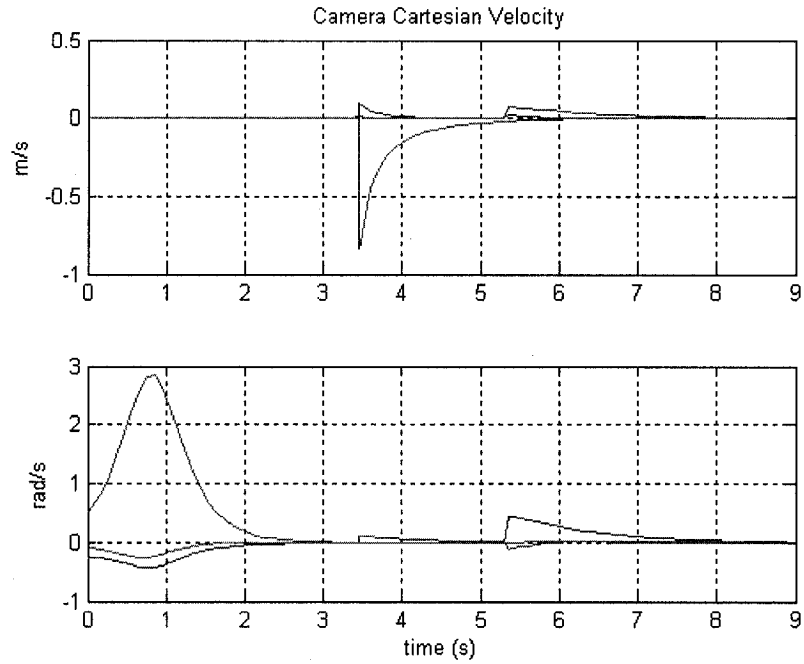
Figure 5-2 Simulation results of task 2

From Figure 5-2 (e), we can notice that the end effector trajectory in Cartesian space is closed to a straight line. In Figure 5-2 (b), it is easy to notice that the camera velocity in Cartesian space consists of three segments corresponding three stages of the control algorithm.

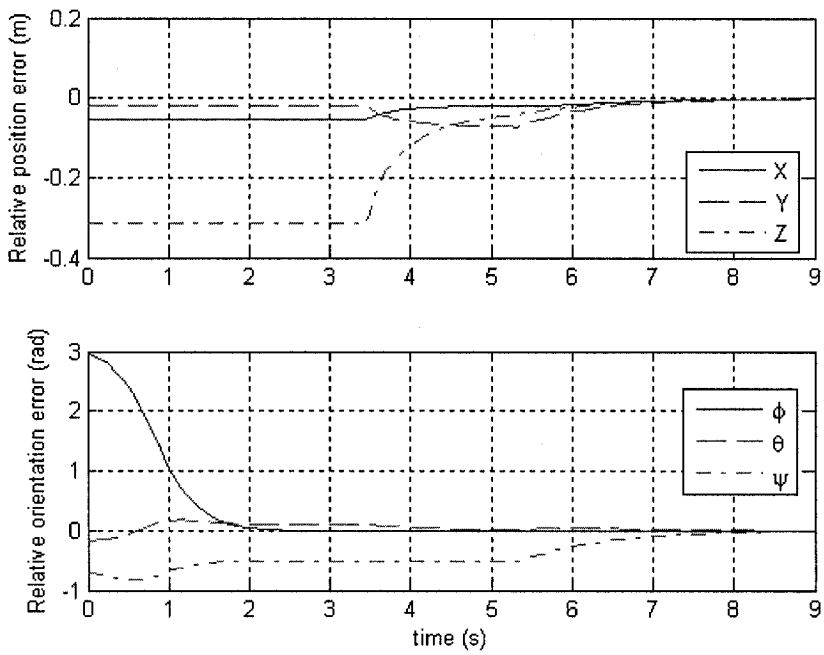
The third task is with a relative large pose discrepancy, in which the traditional IBVS is failed. As shown in Figure 5-3 (c), the relative position errors and orientation errors are up to 0.4m and 3 rad respectively.



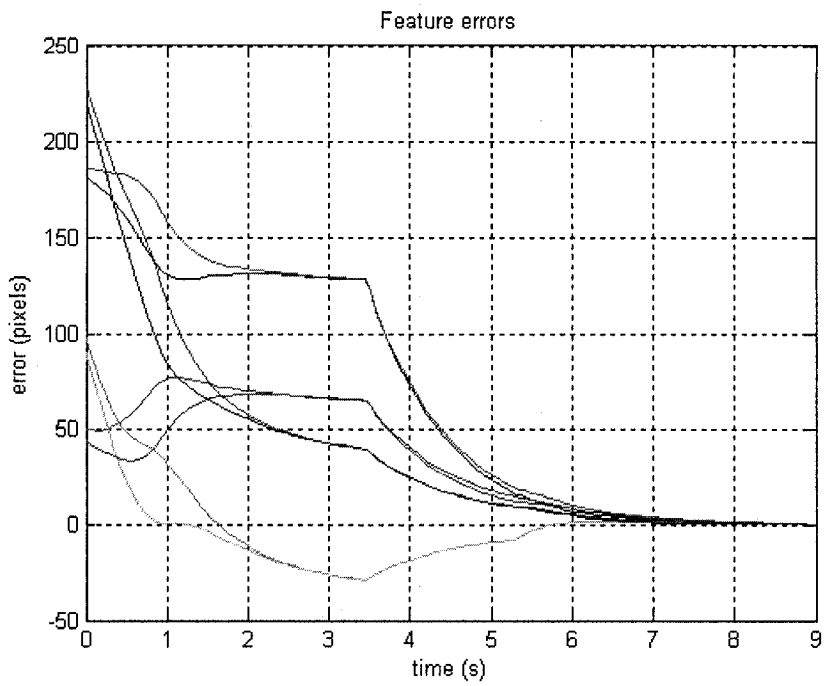
(a) Image trajectory



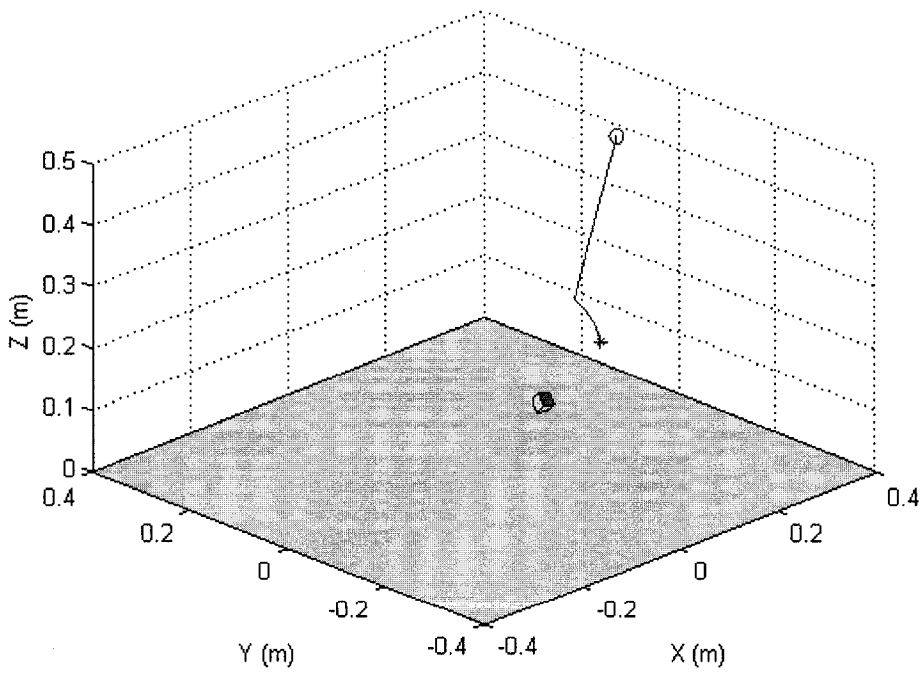
(b) Camera translational and rotational velocity in Cartesian space



(c) Relative position error and orientation error of robot end effector



(d) Image feature errors



(e) End effector trajectory in Cartesian space

Figure 5-3 Simulation results of task 3

The proposed method is successful for task 3, whose pose discrepancy is relative large. From Figure 5-3 (b), notice that some image features errors increase during the first stage. The reason is that the first stage's control goal is to minimize the image features errors with respect to imaginary desired features instead of the finally desired image features. Therefore, to project the laser spot on the object will sometimes increase some image features errors.

Through task 1 to task 3, all the object features are supposed to be located in the field of view. However it is not all the cases. Sometimes only partial object in the field of view and part of object features are lost. The task 4 is related to this situation, in which all the other methods are failed. The image trajectory of this test is shown in Figure 5-4.

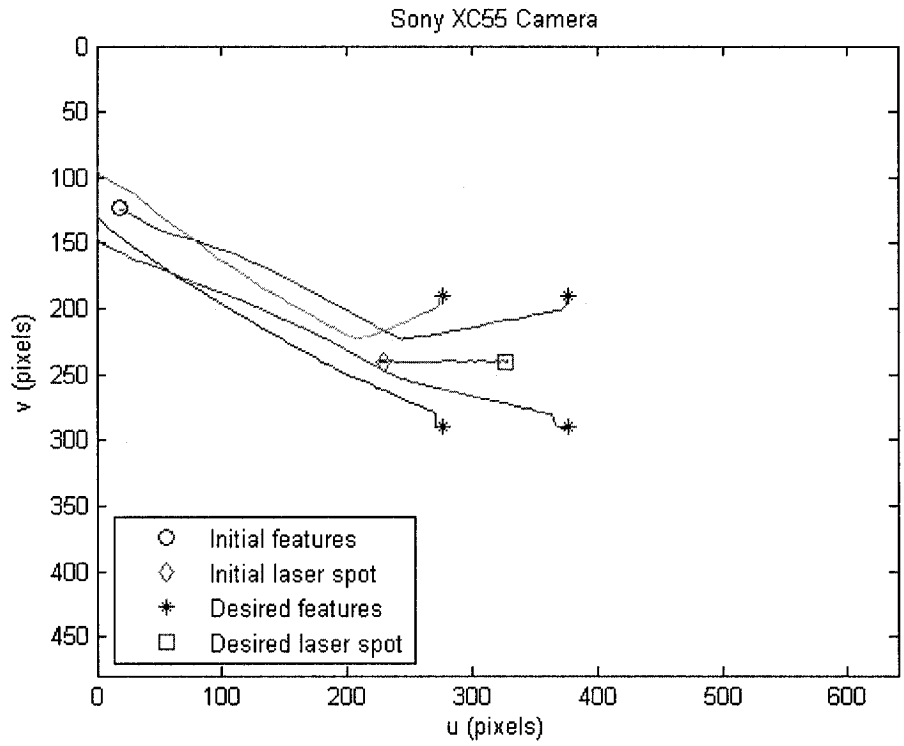


Figure 5-4 Image trajectory of task 4

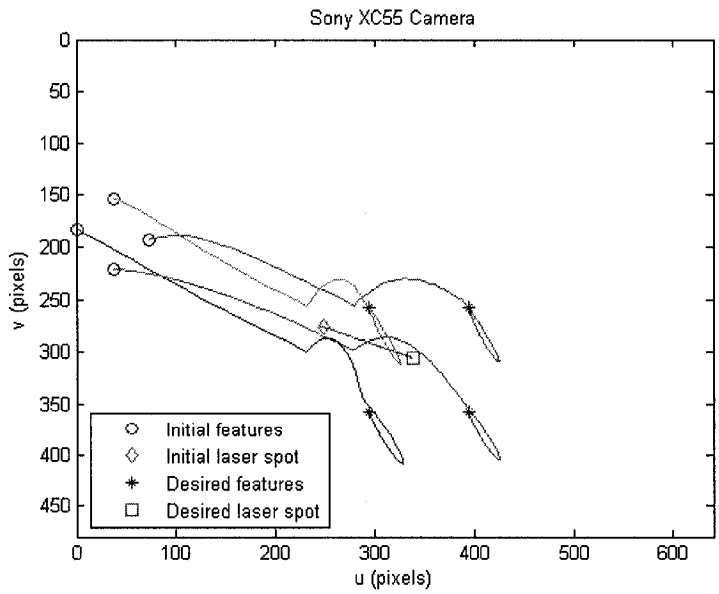
5.2.2 Robustness Test

In this section, the robustness of the proposed algorithm has been tested. Camera calibration error is added to intrinsic parameters with 20% which is shown in Table 5-3. The actual value and applied value of transformation among camera reference, laser pointer frame and robot end effector frame are also shown in Table 5-3.

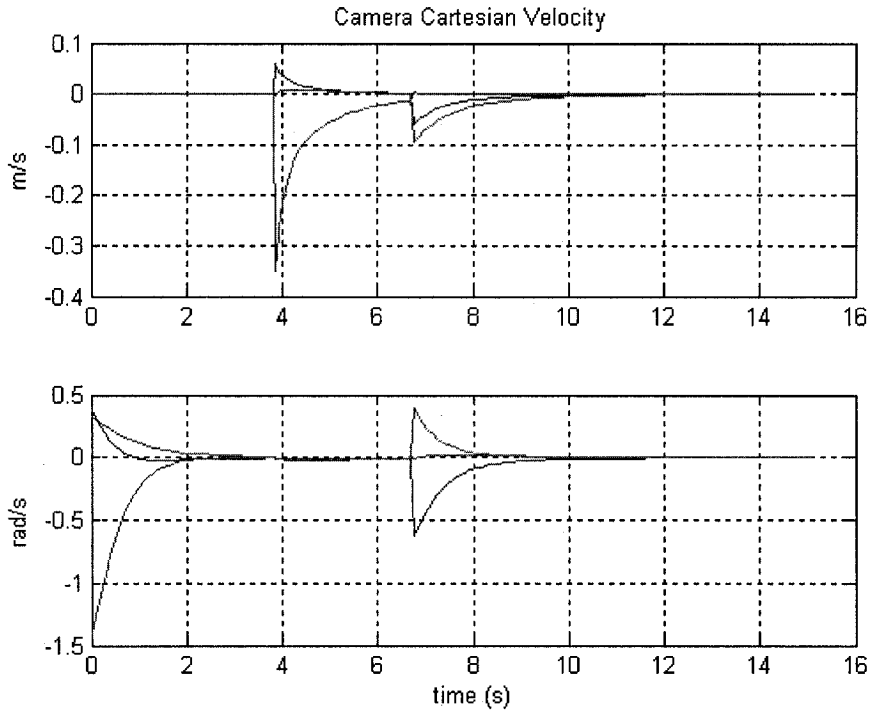
Table 5-3 Actual values and applied values of system parameters

Parameters	Actual value						Applied value					
Principal point (pixel)	[320 240]						[384 288]					
Focal length (mm)	6						7.2					
Effective size (mm)	0.0074×0.0074						0.0074×0.0074					
${}^{C}\mathbf{H}_{L}$ $[X \ Y \ Z \ \varphi \ \theta \ \psi]$ (mm, deg)	30	0	0	0	-10	0	30	10	10	0	-12	2
${}^{E}\mathbf{H}_{C}$ $[X \ Y \ Z \ \varphi \ \theta \ \psi]$ (mm, deg)	-20	20	20	5	5	5	0	0	0	0	0	0

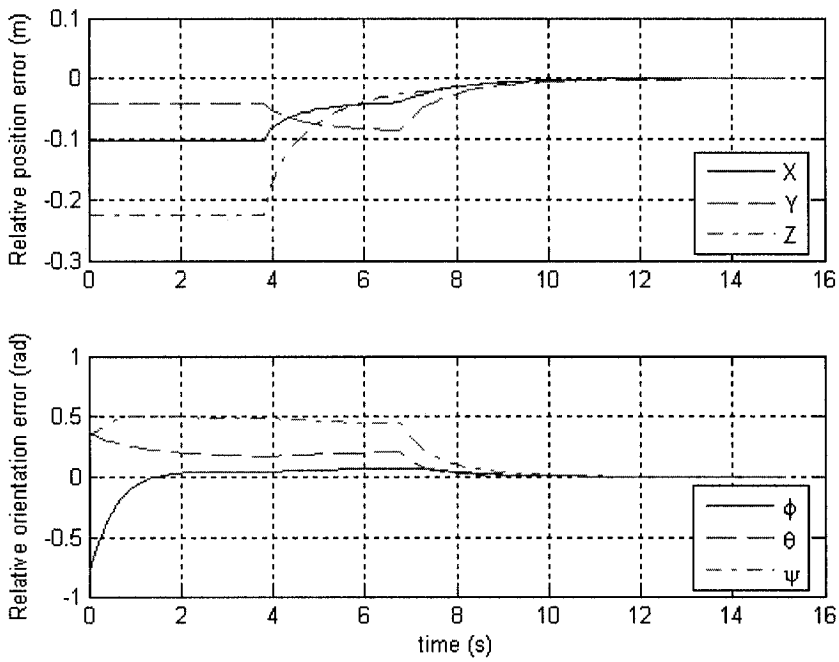
Task 2 in Table 5-1 is chosen for robustness test. The simulation results are shown in Figure 5-5.



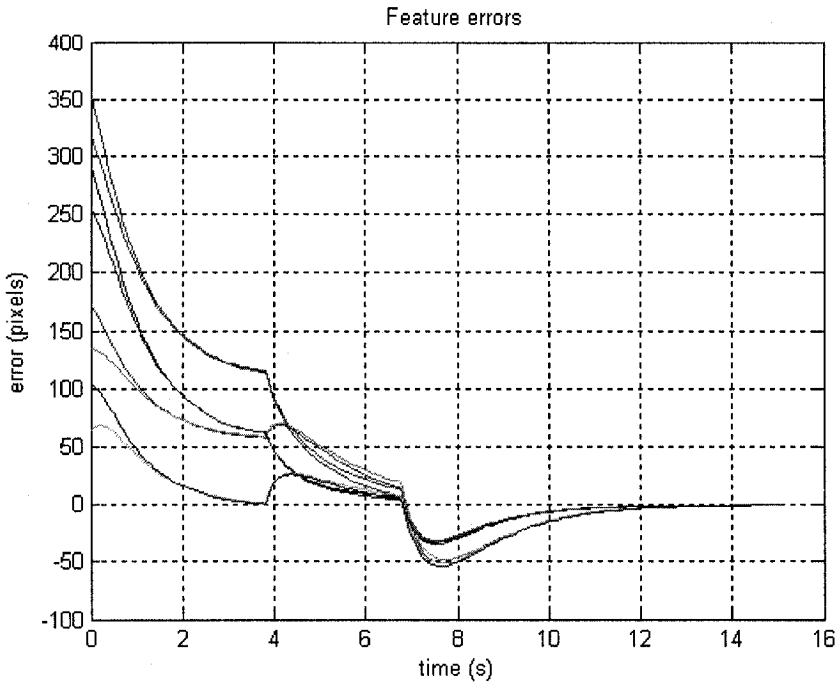
(a) Image trajectory



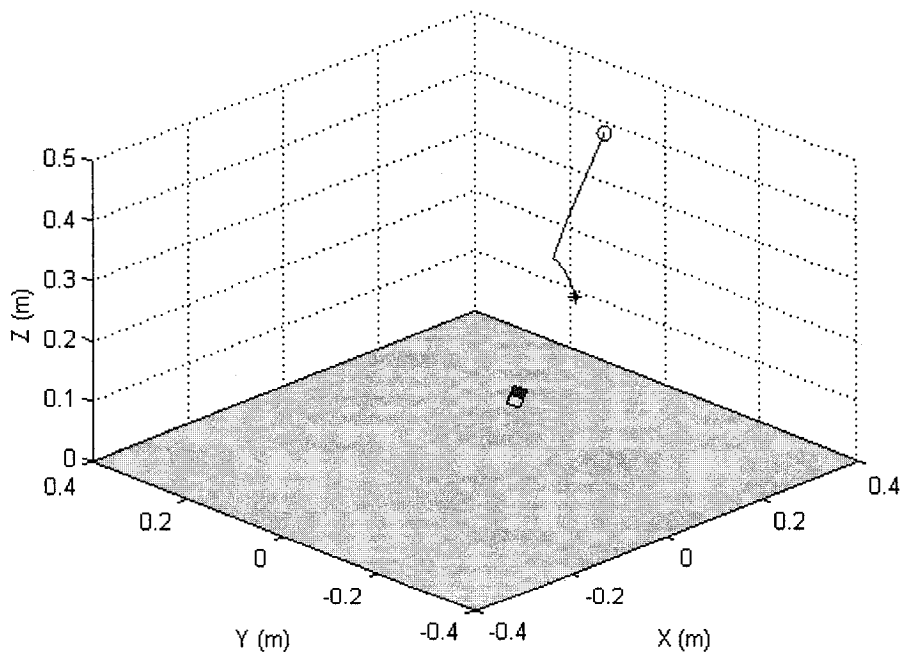
(b) Camera translational and rotational velocity in Cartesian space



(c) Relative position error and orientation error of robot end effector



(d) Image feature errors



(e) End effector trajectory in Cartesian space

Figure 5-5 Robustness test of task 2

As can be seen in Figure 5-5, although the image trajectory is distorted and robot end effector trajectory is lengthened, the task is still able to be achieved. The proposed algorithm is very robust to the camera calibration and hand-eye calibration error as traditional IBVS. However, it is noted that because of bad calibration of camera-laser configuration, the laser trajectory in Figure 5-5 (a) is not a line parallel with camera u axis (see Figure 5-2 a for example) any more. Therefore, the proportional gain K_2 in (4-49) needs to be increased to keep the laser spot focusing on the same point on the target surface.

5.2.3 Comparison

Tasks in Table 5-1 are chosen to compare four different methods, which are proposed IBVS with laser, 2.5D, traditional IBVS and PBVS. For traditional IBVS, on-line updating image Jacobian method is applied with exact depth for each point.

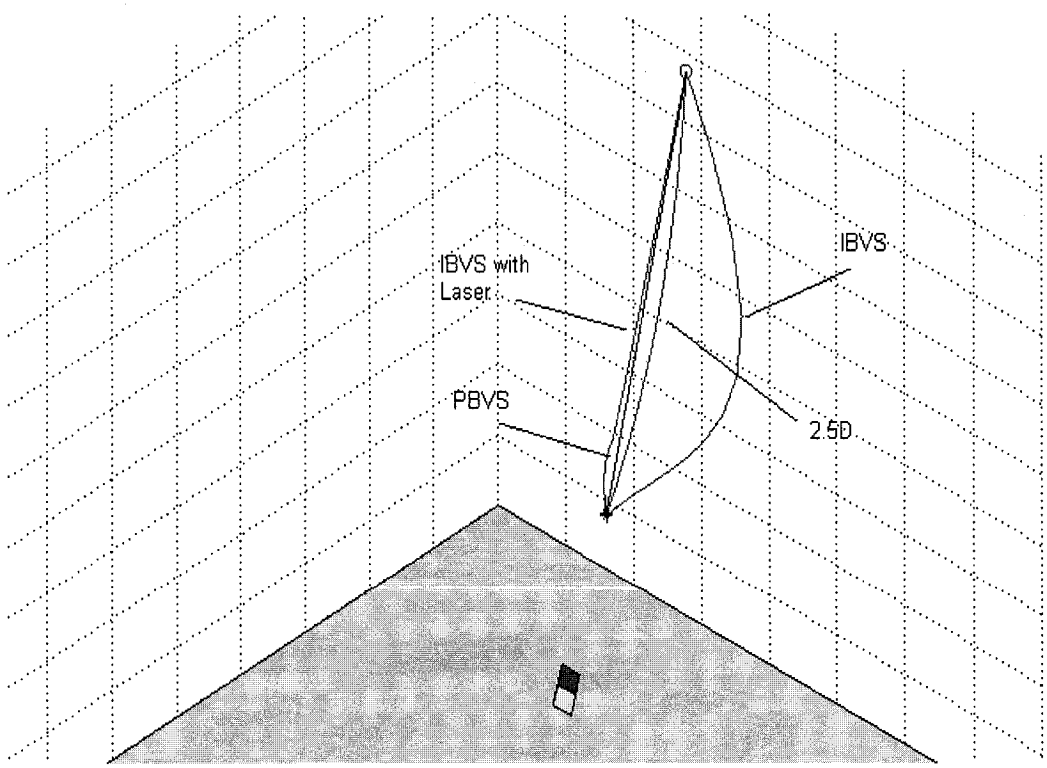


Figure 5-6 End effector trajectories of task 2 in Cartesian space

The simulation result of the end effector trajectories of task 2 is shown in Figure 5-6. It is noted that the end effector trajectory of PBVS using DeMenthon's pose estimation method [14] (with four coplanar features and one noncoplanar feature) gives a straight line in Cartesian space. However the image trajectory shown in Figure 5-7 has been out of the field of view. During the simulation, in order to see the result of this PBVS method, we did not stop the servoing when the features are lost the field of view. Minus pixels in v axis of image plane are taken to finish the servoing. In the real situation, the servoing will be failed when the vision feedback is gone.

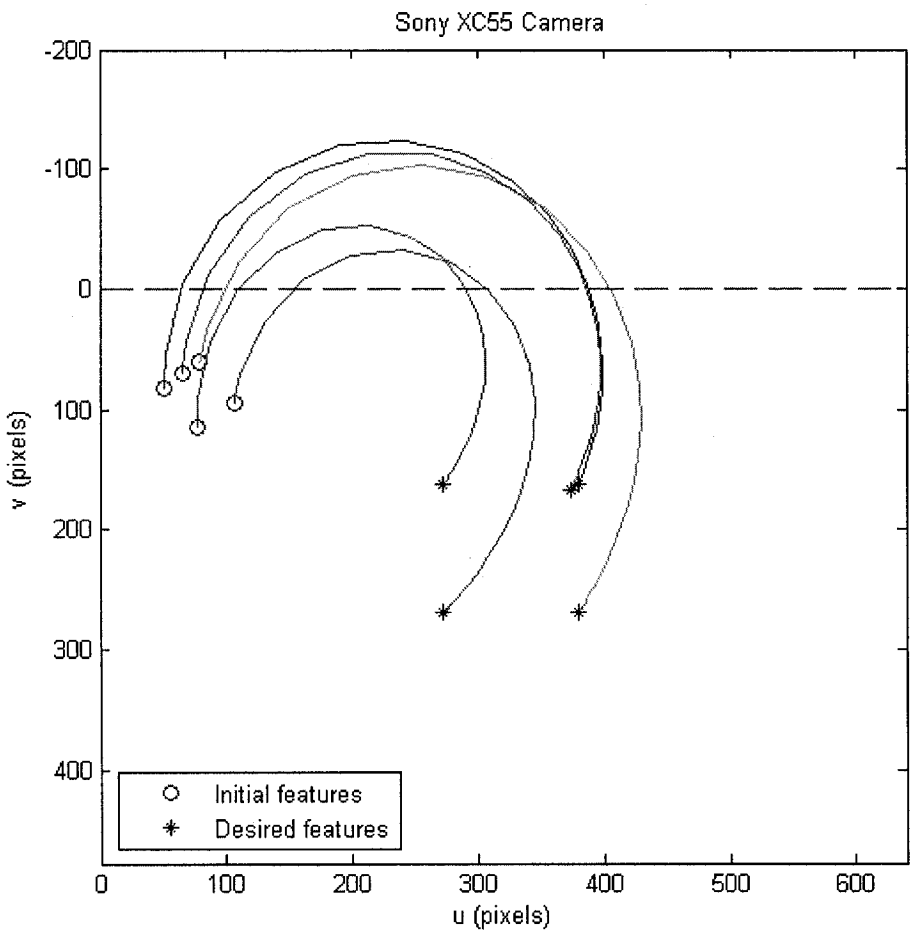


Figure 5-7 Image trajectory of task 2 using PBVS

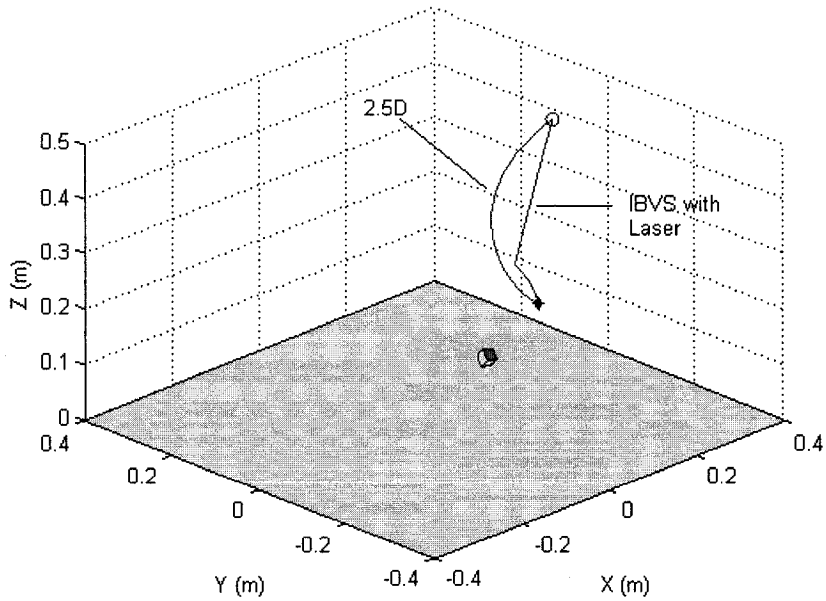


Figure 5-8 End effector trajectories of task 3 in Cartesian space

In Figure 5-8, only the results of IBVS with laser method and 2.5D method are given, since the traditional IBVS fails for task 1 and 3 due to image singularity and large pose discrepancy respectively.

Table 5-4 Length of Cartesian trajectory

	IBVS with Laser (m)	2.5D (m)	IBVS (m)
Task 1	0	0.01	failure
Task 2	0.348	0.308	0.404
Task 3	0.364	0.398	failure
Task 4	0.361	failure	failure

For task 2, the length of Cartesian trajectory of the traditional IBVS is the longest. Comparing IBVS with laser method and 2.5D method, one can see that for task 2, the length of trajectory of the former method is longer than the later method. However, being

opposite for task 3, it is noted that for 2.5D method, when large rotational discrepancy exists, unnecessary motions of end effector are performed.

On the other hand, although the large rotational discrepancy does not affect the length of trajectory for the IBVS with laser method, the trajectory is still under improvement.

5.3 Experimental Results of IBVS with Laser System

In this section, the proposed IBVS with laser system has been tested on a robotic manufacturing system. The detailed description has been presented in Table 3-3. The good and bad calibration values are shown in Table 5-5. The d and α for the fixed laser-camera configuration are set to 8mm and 72deg respectively.

Table 5-5 Good and bad calibration values of system parameters

Parameters	Good calibration values						Bad calibration values					
Principal point (pixel)	[326 228]						[384 288]					
Focal length (mm)	6.02						7.2					
Effective size (mm)	0.0074×0.0074						0.0074×0.0074					
${}^{C}\mathbf{H}_{L}\ [X\ \ Y\ \ Z\ \ \varphi\ \ \theta\ \ \psi]$ (mm, deg) Actual value is measured by hand	8	0	0	0	-18	0	6	0	0	0	-14	0
${}^{E}\mathbf{H}_{C}\ [X\ \ Y\ \ Z\ \ \varphi\ \ \theta\ \ \psi]$ (mm, deg) Actual value is calculated by calibration	-7	5	189	-2	0	-179	-20	20	210	5	5	180

A commercial laser pointer FAY-IS3-1RD [80] is shown in Figure 5-9.

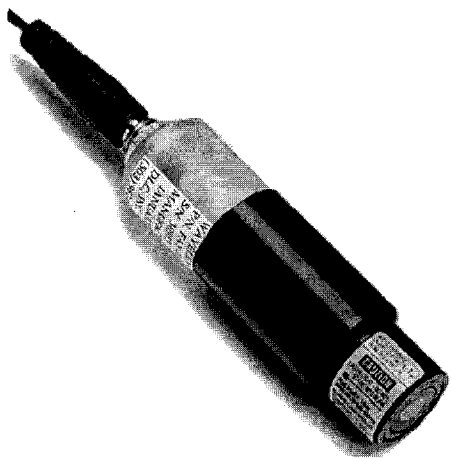


Figure 5-9 Laser pointer

The robotic manufacturing system setup for testing IBVS with laser pointer is shown in Figure 5-10.

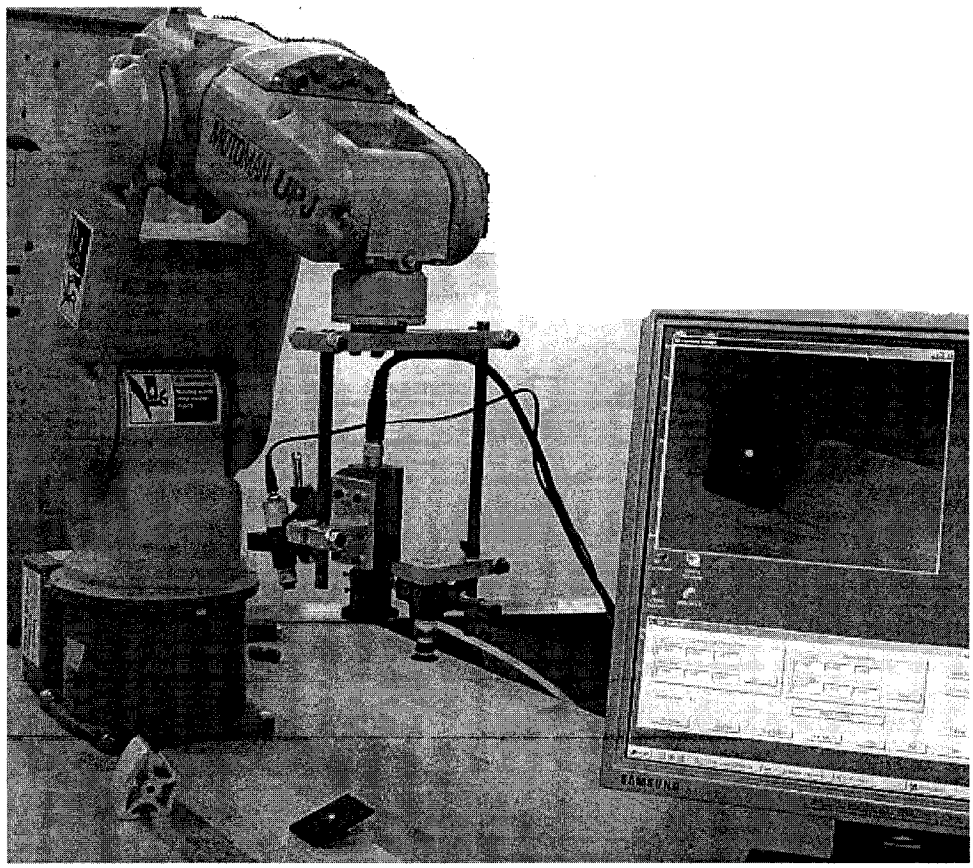


Figure 5-10 Robotic manufacturing system setup for IBVS with laser pointer

To verify the effectiveness of the proposed method, a plastic object shown in Figure 5-11 is chosen to be assembled in a metallic part. The four coplanar corners of the surface are selected as the targets.

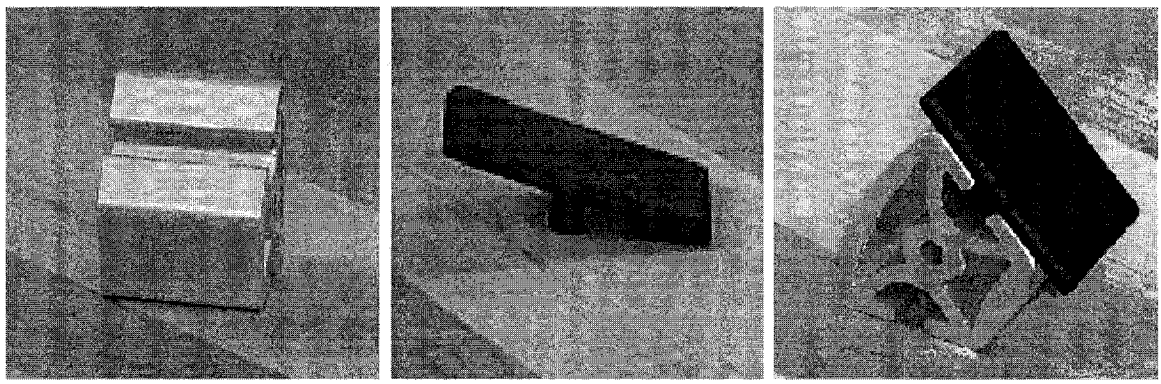


Figure 5-11 Components to be assembled

The detailed experimental setup is described in Table 3-3 and the diagram of the experimental system is presented in Figure 3-7. The robot controller for UPJ robot is JRC. One of the advantages of JRC controller is that it accepts the position and orientation values and calculates the joint angle itself, which avoids the robot kinematic modeling error. However, the drawback of the JRC controller is that it cannot be used for real time control. In other words, during each iteration, when the designed IBVS with laser controller generates a new value and send it to JRC controller, it will not respond it until the prior position is reached. With this limitation of hardware, we have to divide the calculated value of each iteration into a small value by a constant factor, and increase the sampling time as well. In our experiment, we choose that constant factor as fifty, and the sampling time as two seconds. Also due to this limitation we cannot present the camera velocity curve in Cartesian space. The robotic assembly procedures are presented in Figure C-3.

The desired pose ${}^{E}\mathbf{H}_{O}$ is set to $[36 \quad -66 \quad 123 \quad 0 \quad 0 \quad 13.7]$ (mm, deg) by teaching procedure presented in Figure C-2. The desired image shown in Figure 5-12 is taken at this relative pose.

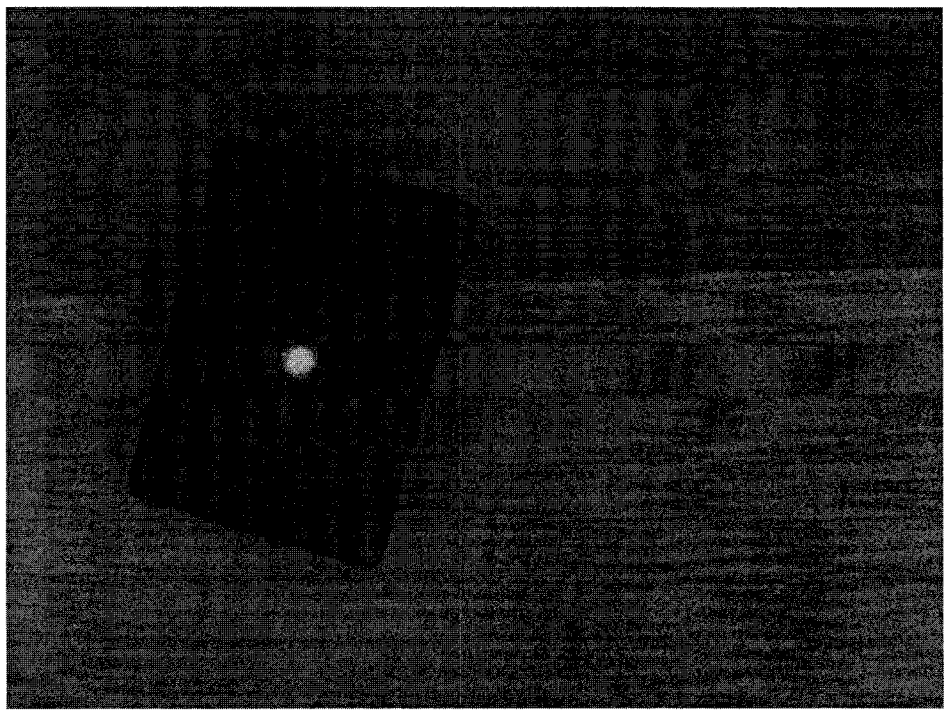


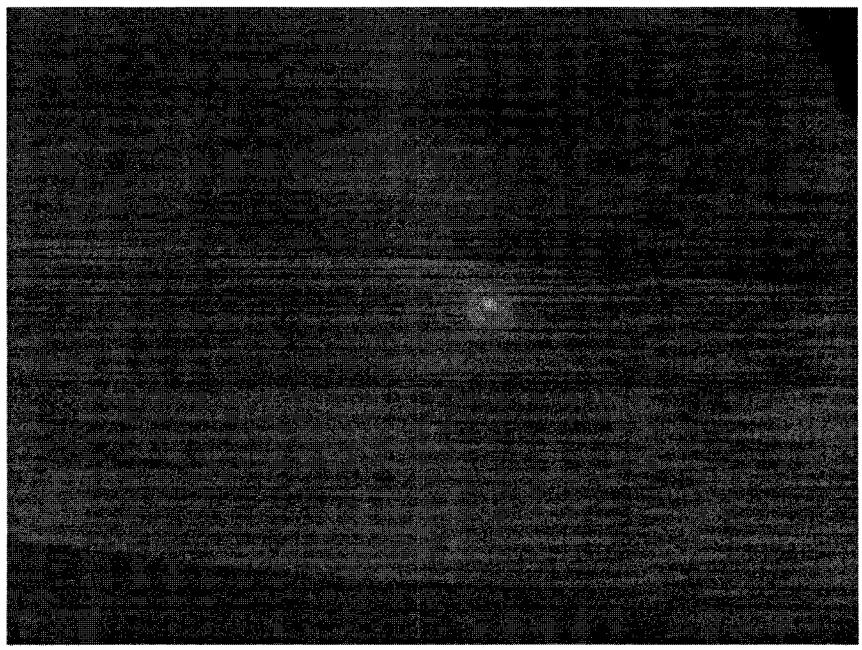
Figure 5-12 Desired image

The predetermined threshold f_0 in (4-27) is set as 10 pixels. The f_0^l in (4-43) is tuned as 10 pixels, which is approximate to 28mm in our experimental setup. All the proportional gains of control law are set to 1.

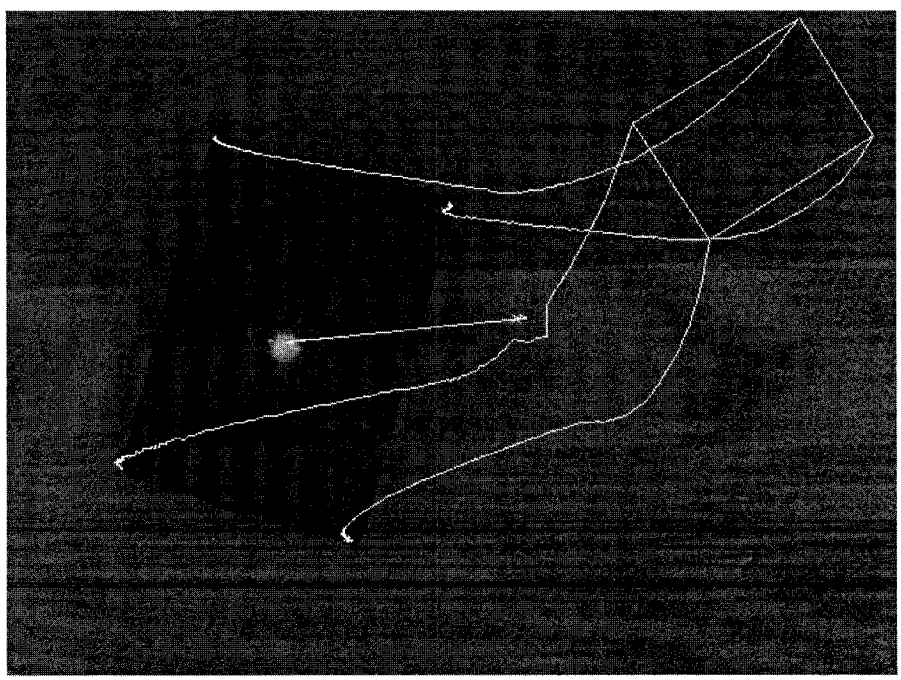
When the condition of feature error norm falling below 3 pixels is achieved, the end effector is brought to the desired pose with respect to the plastic object, which is a fixed transformation for any object position. Therefore, commanding the robot end effector to move that predefined transformation vector in end effector frame will perfectly let the vacuum pump suck up the object. The assembled components shown in Figure 5-11 present the accuracy of the proposed method.

5.3.1 With Good Calibration Value

Firstly, the test of partial object in the field of view is performed.



(a) Initial image



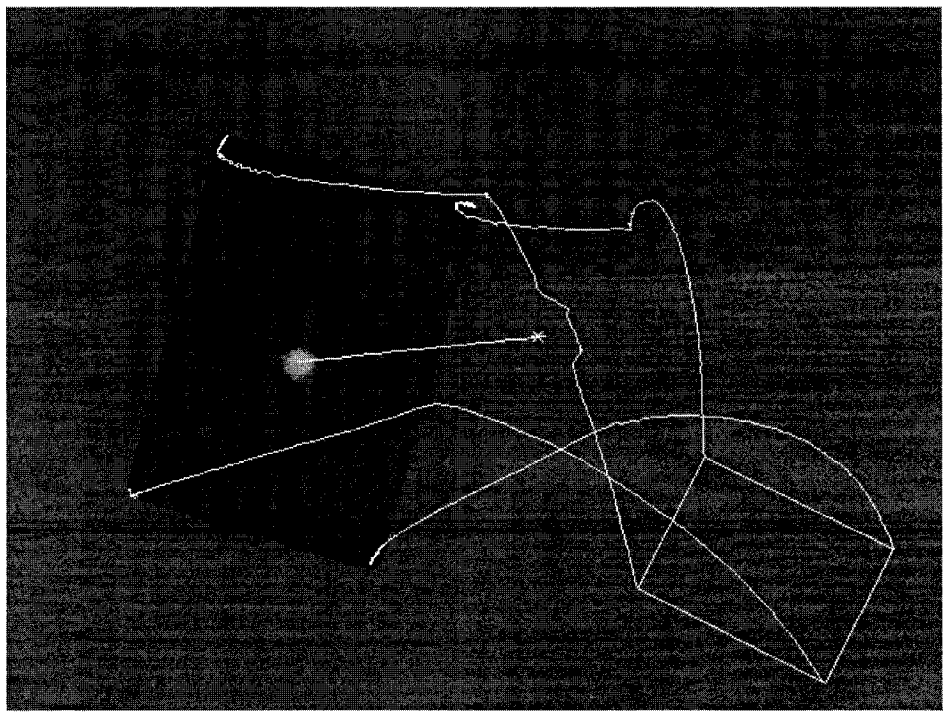
(b) Image trajectory

Figure 5-13 Experimental results of partial object in the Field of View

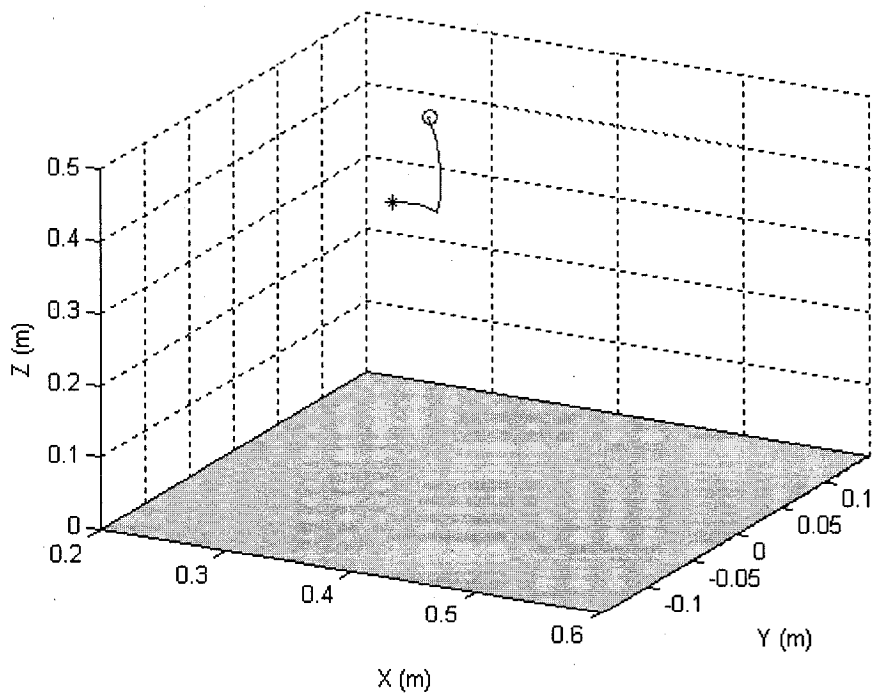
The initial image is demonstrated in Figure 5-13 (a). Only a small part (top right corner) of the object is in the image and all the corner features are unavailable. The image trajectory of this situation is shown in Figure 5-13 (b). Since part of object features are out of the image during the beginning of servoing, we start to draw the trajectory only from all the features are available.

Secondly, experiment of all features in the field of view is carried out. The initial image is presented in Figure 5-14 (a). The object position is random. It can be found that a distortion exist at the right side of the cross sign in Figure 5-14 (b). That is because the laser spot accidentally projects on the object corner and affect the result of corner detection. But it will not affect the eventual result. The end effector trajectory in Cartesian space is given in Figure 5-14 (c).

(a) Initial image



(b) Image trajectory

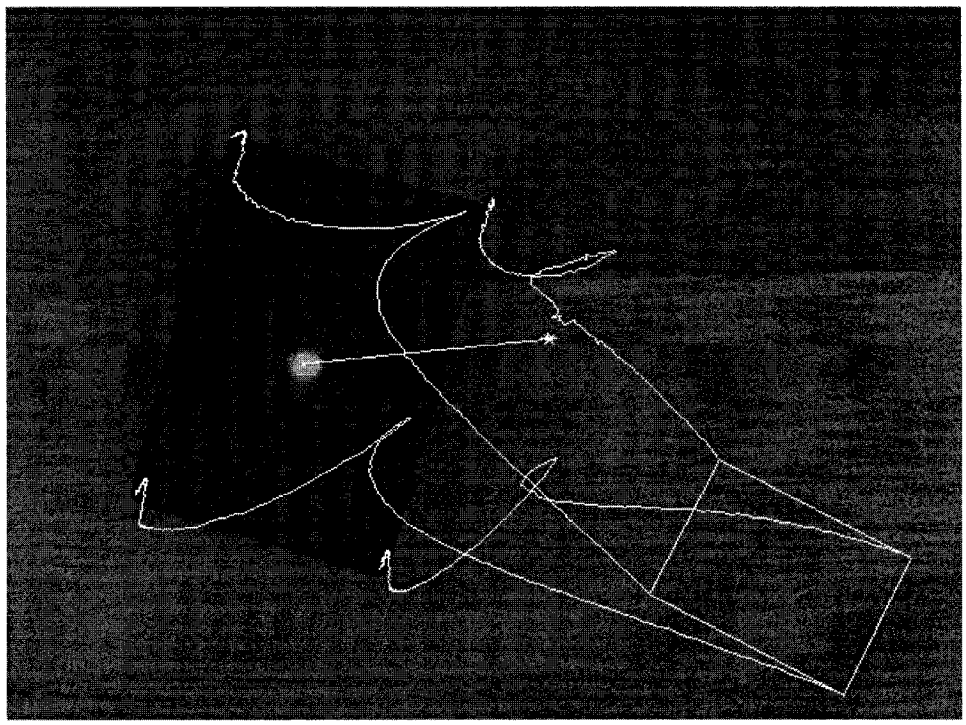


(c) End effector trajectory in Cartesian space

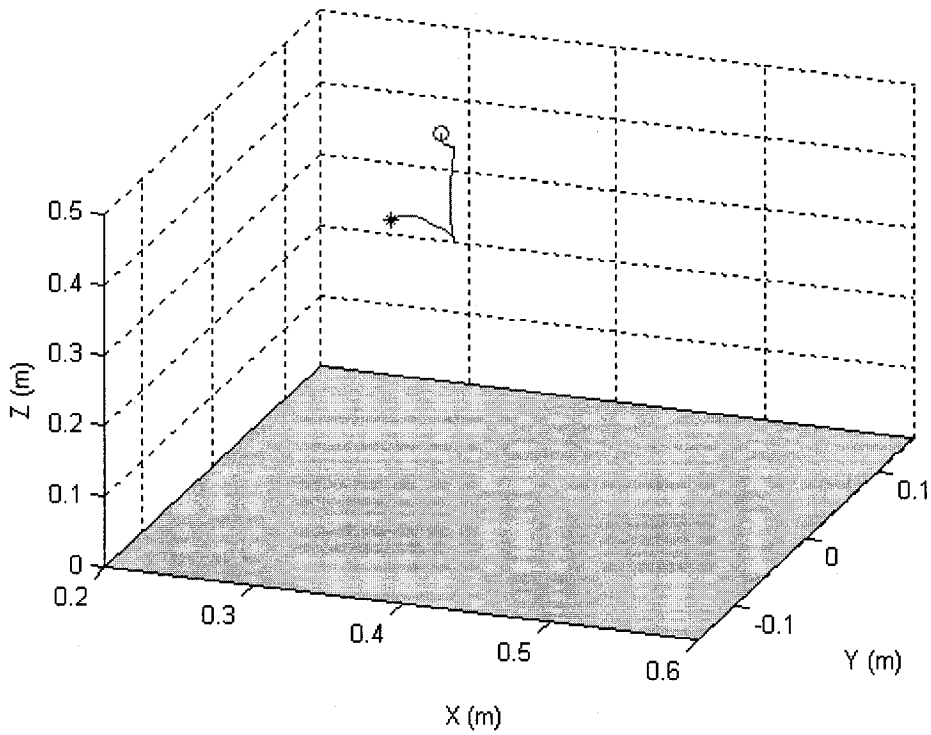
Figure 5-14 Experimental results with good calibration value

5.3.2 With Bad Calibration Value

To test the robustness of the IBVS with laser pointer system, camera calibration error is added to intrinsic parameters with 20% which is shown in Table 5-5. The good calibration value and bad calibration value of transformation among camera reference, laser pointer frame and robot end effector frame are also shown in Table 5-5. The object position is same with the experiment of good calibration value and the initial image is shown in Figure 5-14 (a).



(a) Image trajectory



(b) End effector trajectory in Cartesian space

Figure 5-15 Experimental results with bad calibration value

Although the image trajectory and robot end effector trajectory shown in Figure 5-15 is distorted comparing with the trajectories presented in Figure 5-14, the image features still converge to the desired position and the assembly task is able to be achieved as well. As can be seen in Figure 5-15, the proposed algorithm is very robust to the camera calibration and hand-eye calibration error.

5.4 Summary

In this chapter, first the simulation results of IBVS with laser pointer system are given. Four selected tasks covering most situations are tested. The simulation results show that the new system inherits the advantages, improves the performance, and conquers the weaknesses of the traditional IBVS.

In the second part of this chapter, the proposed IBVS with laser method is experimentally tested in a robotic manufacturing system. Since it is hard to quantify the precision through measuring the relative pose of the object with respect to robot frame in the experimental setup, a plastic object is chosen to be assembled in a metallic part to verify the effectiveness. Both good and bad calibration results are applied to validate the proposed system.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 Introduction

In this chapter, the conclusions and contributions of this research are summarized from the simulation and experimental results and some possible extensions and future work are suggested as well.

6.2 Conclusions

This thesis presents two novel methods for visual servoing in robotic manufactory system. The conclusions and contributions can be summarized from the simulation and experimental results as follows.

1. A novel Neural Network (NN) Based Hand-Eye Calibration is developed.
    - A MultiLayer Perceptron (MLP) NN is used to approximate the highly nonlinear coordinate transform from object image coordinates to real world object coordinates as the camera moves with the robot.
    - The main advantages of NN based hand-eye calibration are that it can solve the hand-eye calibration problem without the estimation of hand-eye transformation and improve the tracking accuracy as well.
    - By choosing some random points in the workspace to simulate a moving object, we verify the effectiveness of the proposed calibration method for continuous object tracking.

- The experimental results show that the proposed calibration method for 2D object tracking outperforms the current solving homogeneous transformation matrix method and free hand-eye calibration method.

2. A new approach to IBVS with laser pointer was developed
   - The proposed method is derived from traditional IBVS. Therefore it inherits the advantages of IBVS, which are robust to the camera calibration and hand-eye calibration, and is object model free as well.
   - Comparing with traditional IBVS, it avoids image singularities and image local minima, and is successful for large pose discrepancy. Furthermore, the trajectory of the robot end effector in Cartesian space is shortened.
   - By pointing the laser spot on the centroid of the object in image plane, the proposed method is successful for the case that only part of the object is in the field of view, in which all the other methods fail.
   - The success of the experiment of IBVS with laser system not only verifies the algorithm but also validates the feasibility of this system to be applied in industrial project.

6.3 Future Work

Although the proposed two novel methods of visual servoing in robotic manufacturing system are effective, some improvements still need to be made in the future work. The following points outline the suggestions.

1. Hand-eye calibration
   - Design a NN to be fit for the situation that the object has 6DOF pose with respect to the camera.

- Apply some other intelligent algorithms to solve the hand-eye calibration problem, such as machine learning using Support Vector Machine (SVM).

2. IBVS with laser system
   - Test the system on a robot manipulator supporting real time control, so that a shorter sampling time could be applied to increase the real time ability and reduce the whole processing time to fulfill the task.
   - Improve the robot end effector trajectory in Cartesian space to be more closed to a straight line as PBVS.
   - During the experiment, the precision of assembly is partially based on the accuracy of corner detection. Sometimes point feature is easy to be disturbed by image noise. Therefore, some additional features are better to be chosen to improve the robustness of image noise.

REFERENCES

[1] N. Aouf, H. Rajabi,, N. Rajabi, H. Alanbari and C. Perron, "Visual object tracking by a camera mounted on a 6DOF industrial robot," IEEE Conference on Robotics, Automation and Mechatronics, Vol. 1, 1-3 pp. 213-218, Dec. 2004.

[2] http://www.robots.com/motoman-up6-pics.htm

[3] http://www.robots.com/robot-drilling.htm

[4] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," IEEE Transactions on Robotics and Automation, vol. 12, pp. 651–670, Oct. 1996.

[5] P. I. Corke, "Visual Control of Robots: High Performance Visual Servoing," Research Studies Press, Australia, 1996.

[6] P. I. Corke and S A. Hutchinson, "Real-Time Vision, Tracking and Control," in Proceedings ICRA '00 IEEE International Conference on Robotics and Automation, vol. 1, pp. 622-629, 2000

[7] Motoman UPJ Reference Manual, MOTOMAN INC., 2003

[8] http://www.motoman.com/products/datasheets/UPJ.pdf

[9] J. Hill and W. T. Park, "Real time control of a robot with a mobile camera," in proceeding 9th ISIR, pp. 233-246,1979

[10] G. Chesi and K. Hashimoto, "Static-eye against hand-eye visual servoing," in Proceedings of the 41st IEEE Conference on Decision and Control, vol. 3, pp. 2854-2859, 2002

[11] G. Flandin, F. Chaumette and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," in Proceedings. ICRA '00 IEEE International Conference on Robotics and Automation, vol. 3, pp. 2741-2746, 2000

[12] A. C. Sanderson, L. E. Weiss, "Image Based Visual Servo Control Using Relational Graph Error Signal", IEEE International Conference of Cybernetics and Society, pp. 1074-1077, 1980

[13] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor based control of robots with visual feedback," IEEE Journal of Robotics and Automation, vol. 3(5), pp. 404–417, Oct. 1987.

[14] D. F. DeMenthon and L.S. Davis, "Model-based object pose in 25 lines of code," International Journal of Computer Vision, vol. 15, no. 1/2, pp. 123-142, 1995.

[15] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative end-effector control using Cartesian position-based visual servoing," IEEE Transactions on Robotics and Automation, vol. 12, pp. 684–696, Oct. 1996.

[16] F. Chaumette, "Potential problems of stability and convergence in image based and position-based visual servoing," in The Confluence of Vision and Control, D. Kriegman, G. D. Hager, and A. Morse, Eds. Berlin, Germany: Springer-Verlag, vol. 237, pp. 66-78. Lecture Notes in Control and Information Sciences, 1998.

[17] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," IEEE Transactions on Robotics and Automation, vol. 8(3), pp. 313–326, 1992.

[18] K. Hashimoto, T. Kimoto, T. Ebine and H. Kimura, "Manipulator Control with Image-Based Visual Servo," in Proceedings IEEE International Conference on Robotics and Automation, vol. 3, pp. 2267 - 2271, April 1991

[19] H. Kase, N. Maru, A. Nishikawa, S. Yamada and F. Miyazaki, "Visual Servoing of the Manipulator using the Stereo Vision," in Proceedings of the IECON '93 IEEE International Conference on Industrial Electronics, Control, and Instrumentation, vol. 3, pp.1791 – 1796, 1993

[20] R. Mahony, P. Corke and F. Chaumette, "Choice of image features for depth-axis control in image based visual servo control," IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.1, pp. 390–395, 2002.

[21] P. Y. Oh, and P. K. Allen, "Visual Servoing by Partitioning Degrees of Freedom," IEEE Transactions on Robotics and Automation, vol. 17, pp. 1-17, 2001

[22] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," IEEE Transactions on Robotics and Automation, vol. 17(4), pp. 507-515, Aug. 2001.

[23] M. Iwatsuki and N. A. Okiyama, "A new formulation of visual servoing based on cylindrical coordinate system," IEEE Transactions on Robotics, (see also IEEE Transactions on Robotics and Automation), vol. 21(2), pp. 266–273, April 2005.

[24] J. Pages, C. Collewet, F. Chaumette and J. Salvi, "Plane-to-plane positioning from image-based visual servoing and structured light," in Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 1004–1009, 2004.

[25] A. Krupa, J. Gangloff, C. Doignon, M. Mathelin. G. Morel, J. Leroy, L. Soler and J. Marescaux, "Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing," IEEE Transaction on Robotic and Automation, vol. 19(5), pp. 842-853, 2003.

[26] A. Krupa, C. Doignon, J. Gangloff and M. Mathelin, "Combined image-based and depth visual servoing applied to robotized laparoscopic surgery," IEEE/RSJ International Conference on Intelligent Robots and System, vol. 1, pp. 323 – 329, 2002.

[27] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," IEEE Transactions on Robotics and Automation, vol. 18(2), pp. 176–186, Apr. 2002.

[28] E. Malis, F. Chaumette, and S. Boudet, "2-1/2-D visual servoing," IEEE Transactions on Robotics and Automation, vol. 15(2), pp. 238–250, Apr. 1999.

[29] F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings," in Proceedings ICRA '00 IEEE International Conference on Robotics and Automation, vol. 1, pp. 630-635, April 2000

[30] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," International Journal of Pattern Recognition and Artificial Intelligence, vol. 2-3, 1988

[31] O. Faugeras, "Three-Dimensional Computer Vision," Cambridge, MA: MIT Press, 1993

[32] N. R. Gans and S. A. Hutchinson, "A switching approach to visual servo control," in Proceedings, IEEE International Symposium on Intelligent Control, pp. 770–776, 2002.

[33] N. R. Gans and S. A. Hutchinson, "An experimental study of hybrid switched system approaches to visual servoing," in Proceedings ICRA '03, IEEE International Conference on Robotics and Automation, vol. 3, pp. 3061–3068, Sept. 2003.

[34] N. R. Gans, S. A. Hutchinson and P. I. Corke, "Performance Tests for Visual Servo Control Systems, with Application to Partitioned Approaches to Visual Servo Control," The International Journal of Robotics Research, vol. 22: pp. 955 - 981. 10 2003

[35] N. R. Gans, P. I. Corke and S. A. Hutchinson, "Performance tests of partitioned approaches to visual servo control" in Proceedings. ICRA '02 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1616-1623, 2002

[36] L. Deng, F. Janabi-Sharifi, and W. J. Wilson, "Hybrid strategies for image constraints avoidance in visual servoing," IEEE/RSJ International Conference on Intelligent Robots and System, vol. 1, pp. 348-353, 2002.

[37] L. Deng; W. J. Wilson and F. Janabi-Sharifi, "Dynamic performance of the position-based visual servoing method in the cartesian and image spaces," in Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 510–515, 2003.

[38] L. Deng, F. Janabi-Sharifi and W. J. Wilson, "Hybrid Motion Control and Planning Strategies for Visual Servoing," IEEE Transactions on Industrial Electronics, vol. 52(4), pp. 1024–1040, Aug. 2005.

[39] L. Deng, "Comparison of Image-Based and Position-Based Robot Visual Servoing Methods and Improvements," a PhD dissertation of Electrical and Computer Engineering of University of Waterloo, 2003.

[40] S. Remy, M. Dhome, J. M. Lavest and N. Daucher, "Hand-Eye Calibration," in Proceedings of IROS '97 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 1057-1065, 1997

[41] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, pp. 323-344, Aug. 1987.

[42] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," IEEE Transactions on Robotics and Automation, Vol. 5, No.3, pp. 345-358, Jun. 1989.

[43] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB," IEEE Transactions on Robotics and Automation, Vol. 5, No. 1, pp. 16-29, Feb. 1989.

[44] Y. Motai and A. Kosaka, "SmartView: hand-eye robotic calibration for active viewpoint generation and object grasping, " in Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Vol 3, pp. 2183-2190, May 2001.

[45] F. Dornaika and R. Horaud, "Simultaneous Robot-World and Hand-Eye Calibration," IEEE Transaction on Robotics and Automation, Vol. 14, No. 4, pp. 617-622, Aug. 1998.

[46] E. K. P. Chong and S. H. Zak, "An Introduction to Optimization," John Wiley & Sons, Inc., New York, 1996.

[47] Z. Zhang, "A flexible new technique for camera calibration, " IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.

[48] Z. Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations," International Conference on Computer Vision (ICCV'99), Corfu, Greece, pp. 666-673, September 1999.

[49] M. T. Ahmed, E. E. Hemayed and A. A. Farag, "Neurocalibration: a neural network that can tell camera calibration parameters, " in Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol 1, 20-27 pp. 463-468, Sept. 1999.

[50] F. Angrilli, S. Bastianello and R. Da Forno, "Calibration of stereo vision systems by neural networks," in Proceedings IMTC-96 of Instrumentation and Measurement Technology IEEE Conference, "Quality Measurements: The Indispensable Bridge between Theory and Reality", vol. 2, pp. 839 – 842, 1996

[51] D. Yongtae, "Application of neural networks for stereo-camera calibration," International Joint Conference on Neural Networks, IJCNN '99. vol. 4, pp. 2719 – 2722, July 1999

[52] J. Wen and G. Schweitzer, "Hybrid calibration of CCD cameras using artificial neural nets," IEEE International Joint Conference on Neural Networks, vol. 1, 18-21, pp. 337 – 342, Nov. 1991

[53] Junghee Jun and Choongwon Kim, "Robust Camera Calibration Using Neural Network," in Proceedings of the IEEE Region 10 Conference, TENCON 99. vol. 1, 15-17, pp. 694-697, Sept. 1999.

[54] J. Su, Y. Xi, U.D.Hanebeck and G.Schmidt, "Nonlinear visual mapping model for 3-D visual tracking with uncalibrated eye-in-hand robotic system," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 34, No.1, pp. 652 - 659, Feb. 2004.

[55] G. Q. Wei and G. Hirzinger, "Multisensory Visual Servoing by a Neural Network," IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, vol. 29, No.2, pp. 276-280, Apr. 1999.

[56] Z. Li, W. F. Xie and N. Aouf, "A Neural Network Based Hand-Eye Calibration Approach in Robotic Manufacturing Systems", CSME 2006, Calgary, May 2006.

[57] E. Trucco and A. Verri "Introductory Techniques for 3-D computer Vision," Prentice Hall; 1998

[58] M. T. Hagan and H. B. Demuth, "Neural Network Design," PWS Publishing Company, US, 1996.

[59] M. T. Hagan, and M.B. Menhaj, "Training feedforward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, Vol. 5, No. 6, pp. 989-993, Nov. 1994.

[60] S. Haykin, "Neural Networks, a Comprehensive Foundation," Prentice Hall Inc., 2ed., US, 1999.

[61] K. Hashimoto and T. Noritsugu, "Enlargement of stable region in visual servo," in Proceedings, the 39th IEEE Conference on Decision and Control, vol. 4, pp. 3927–3932, 2000.

[62] K. Hashimoto and T. Noritsugu, "Potential problems and switching control for visual servoing," in Proceedings, IEEE/RSJ International Conference on Intelligent Robots and System, vol. 1, pp. 423–428, Oct. 2000

[63] W. Chang and M. Chai, "Real-Time Vision-Based Contour Following with Laser Pointer," in Proceedings ICRA '03, IEEE International Conference on Robotics and Automation, vol. 2, pp. 2549–2554, Sept. 2003.

[64] E. Marchand, F. Spindler and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills" IEEE Robotics & Automation Magazine, vol. 12-4, pp. 40-52, 2005

[65] G. Chesi, K. Hashimoto, D. Prattichizzo and A. Vicino, "Keeping Features in the Field of View in Eye-In-Hand Visual Servoing: A Switching Approach," IEEE Transactions on Robotics (see also IEEE Transactions on Robotics and Automation), vol. 20-5, pp. 908-914, 2004

[66] J. S. Park and M. J. Chung, "Path Planning With Uncalibrated Stereo Rig for Image-Based Visual Servoing Under Large Pose Discrepancy," IEEE Transactions on Robotics and Automation, vol. 19-2, pp. 250-258, 2003

[67] Y. Mezouar and F. Chaumette, "Design and tracking of desirable trajectories in the image space by integrating mechanical and visibility constraints," in Proceedings

International Conference on Robotics and Automation, vol. 1, pp. 731–736, May 2001.

[68] R. Sharma and S. A. Hutchinson, "Motion perceptibility and its application to active vision-based servo control," IEEE Transactions on Robotics and Automation, vol. 13(4), pp. 607–617, Aug. 1997.

[69] P. I. Corke, "Robotics toolbox for MATLAB," IEEE Transactions on Robotics and Automation, vol. 3(1), pp. 24-32, 1996.

[70] P. I. Corke, "Machine Vision Toolbox," IEEE Robotics and Automation Magazine, 12(4), pp 16-25, 2005

[71] http://www.vision.caltech.edu/bouguetj/calib_doc/. Camera Calibration Toolbox for Matlab reference manual

[72] Neural Network Toolbox for Matlab

[73] H. Demuth and M. Beale, "Neural Network Toolbox User's Guide For Use with MATLAB®. Version 3.0"

[74] R. P. Paul, "Robot manipulators: mathematics, programming, and control. The computer control of robot manipulators," Cambridge, Mass.: MIT Press, 1981

[75] P.G. Ránky and C.Y. Ho. "Robot modeling: control and applications with software," Springer-Verlag, 1985

[76] C.Y. Ho and Jen Sriwattanathamma, "Robot kinematics: symbolic automation and numerical synthesis," Norwood, N.J.: Ablex Pub. Corp., 1990

[77] P. J. McKerrow, "Introduction to robotics," Sydney, Reading, Mass.: Addison-Wesley Pub. Co., 1991

[78] Sony XC55 camera Reference Manual, SONY
http://www.sony.co.jp/Products/ISP/products/model/nontv/XC55.html

[79] Matrox Frame Grabber Reference Manual, Matrox Imaging
http://www.matrox.com/imaging/products/meteor2_mc/home.cfm

[80] Industrial diode laser modules, Diode Laser Concepts, INC.
http://www.diodelaserconcepts.com/Datasheet-HTML/industrial.html

[81] PIAB VGS^{TM} 3010 Vacuum Gripper
http://www.piab.com/

[82] Intel Open Source Computer Vision Library Reference Manuals
http://www.intel.com/technology/computing/opencv/

[83] Sony Camera Adaptor DC-700/700CE Operating Instructions, SONY
http://www.sony.co.jp/Products/ISP/support/catalog/2006_en/Accessories_S.pdf

[84] B.Espiau, "Effect of camera calibration errors on visual servoing in robotics," in Experimental Robotics III: The 3rd International symposium on experimental robotics, Springer-Verlag, Lecture notes in control and information sciences, vol. 200, 1993

Appendix A Robot Model in Matlab Robotics Toolbox

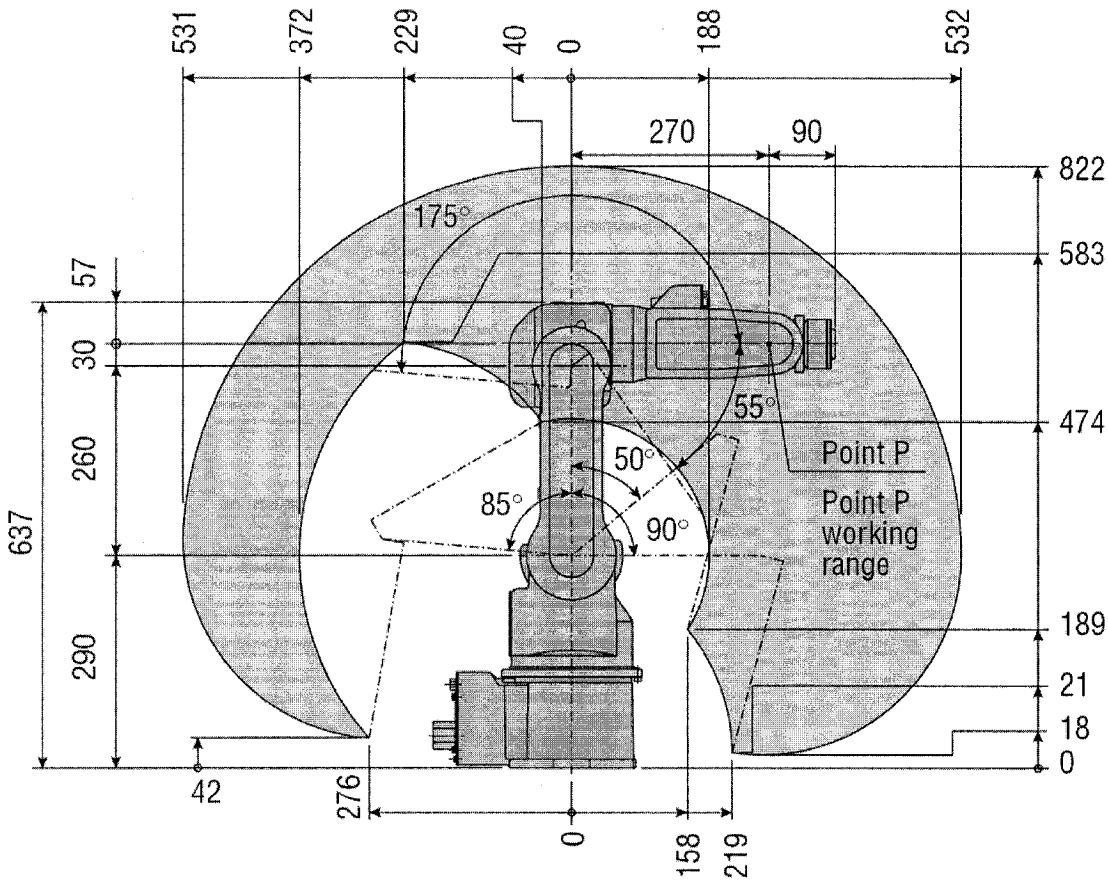


Figure A-1 Motoman UPJ workspace and link dimensions [7]

The Motoman UPJ robot is modeled in Matlab Robotics Toolbox as follows:

```
L{1} = link([pi/2    0    0    0.29    0    0    ], 'standard');
L{2} = link([0       0.26 0    0       0    pi/2 ], 'standard');
L{3} = link([pi/2    0    0    0       0    0    ], 'standard');
L{4} = link([pi/2    0    0    0.27    0    0    ], 'standard');
L{5} = link([-pi/2   0    0    0       0    0    ], 'standard');
L{6} = link([0       0    0    0.09    0    0    ], 'standard');
```

```
L{1}.qlim = [-160 160]* pi/180;

L{2}.qlim = [-85 90 ] * pi/180;

L{3}.qlim = [55 175 ] * pi/180;

L{4}.qlim = [-170 170] * pi/180;

L{5}.qlim = [-120 120] * pi/180;

L{6}.qlim = [-360 360] * pi/180;
```

Appendix B MATLAB Code for NN Training

```
% NNData is a n×6 matrix whose format is [u  v  T_x  T_y  x_W  y_W]
% Input of the network
p=[NNData(:,1)'; NNData(:,2)'; NNData(:,3)'; NNData(:,4)'];
% Outnput of the network
t=[NNData(:,5)'/max(abs(NNData(:,5)')); NNData(:,6)'/max(abs(NNData(:,6)'))];
% Construct the network
net=newff(minmax(p),[20,2],{'tansig','purelin'},'trainlm');
net.trainParam.show = 10;
net.trainParam.lr = 0.03;
net.trainParam.epochs = 500;
% Performance goal
net.trainParam.goal = 1e-8;
% Train the network
[net,tr]=train(net,p,t);
% Simulate the output of the network
[a,Pf,Af,E,perf]=sim(net,p);
Xw=a(1,:).*max(abs(NNData(:,5));
Yw=a(2,:).*max(abs(NNData(:,6));;
```

Appendix C Flow Chart of IBVS with Laser System

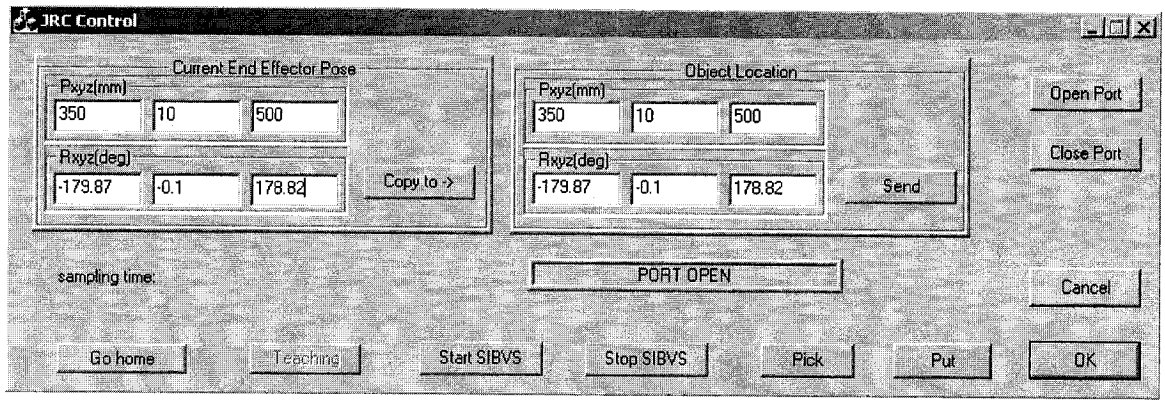


Figure C-1 User interfaces

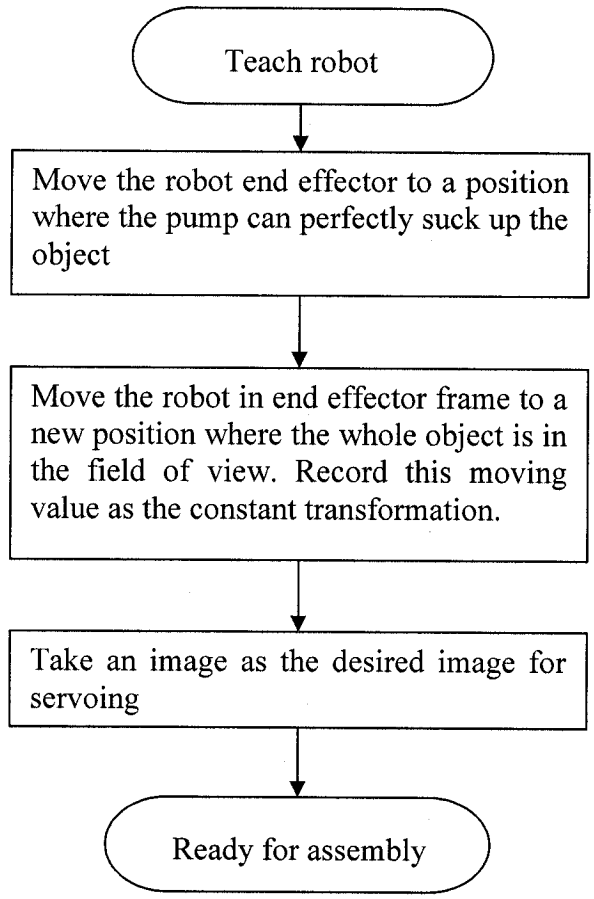


Figure C-2 Teaching procedures

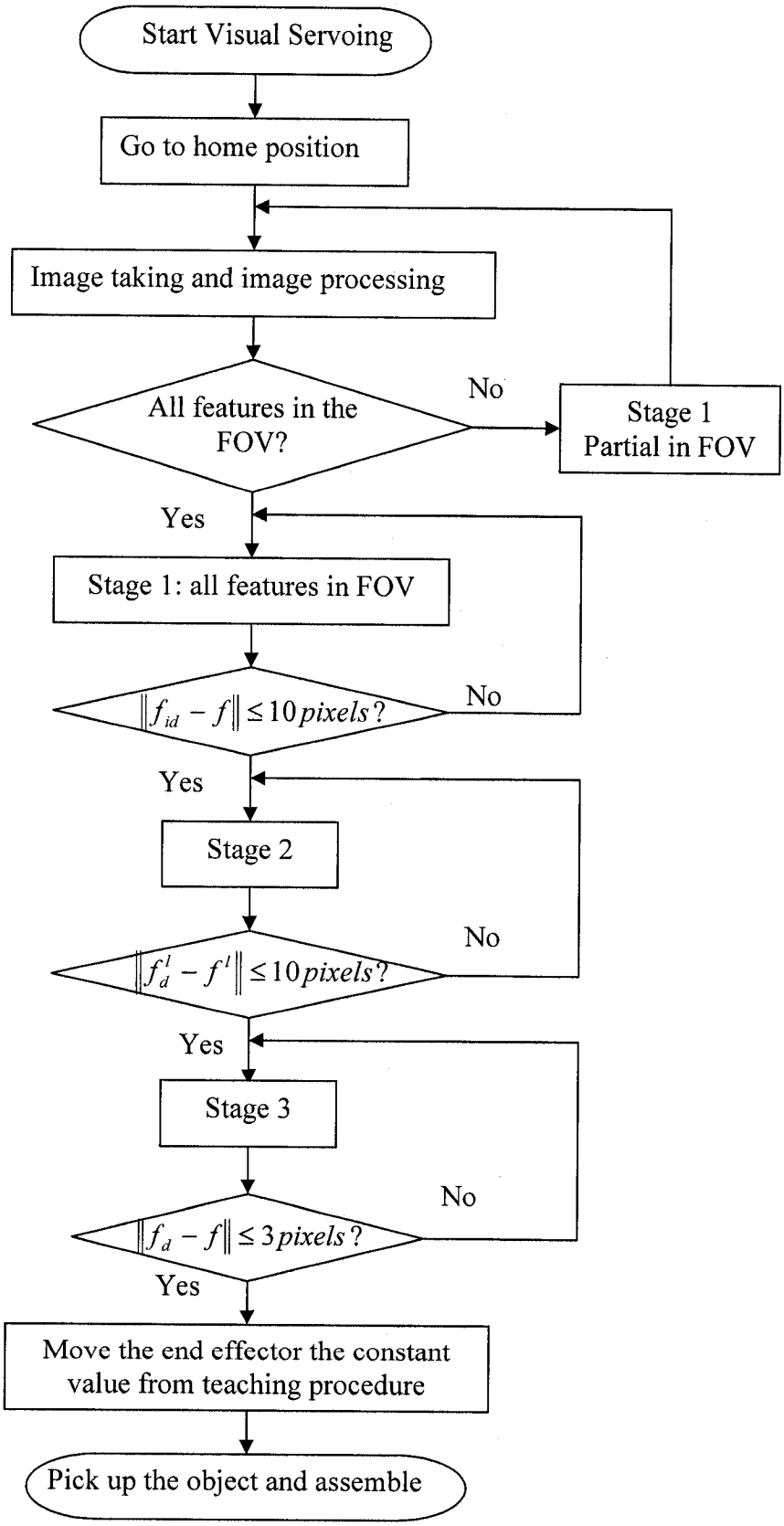


Figure C-3 Assembly procedures