TAMPERE UNIVERSITY OF TECHNOLOGY

YI XIONG

VISION BASED AUTOMATIC CALIBRATION OF MICROROBOTIC SYSTEM

Master of Science Thesis

Examiner: Professor Pasi Kallio
Examiner and topic approved by the Faculty Council of the Faculty of Automation, Mechanical and Materials Engineering on 4th April 2012.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master’s Degree Programme in Machine Automation
XIONG,YI: Vision Based Automatic Calibration of Microrobotic System
Master of Science Thesis, 77 pages, 3 Appendix pages
May 2012
Major subject: Mechatronics
Examiner: Professor Pasi Kallio
Keywords: Calibration, microrobotics, micromanipulation, computer vision, automation

During the last decade, the advancement of microrobotics has provided a powerful tool for micromanipulation in various fields including living cell manipulation, MEMS/MOEMS assembly, and micro-/nanoscale material characterization. Several dexterous micromanipulation systems have been developed and demonstrated. Nowadays, the research on micromanipulation has shifted the scope from the conceptual system development to the industrial applications. Consequently, the future development of this field lies on the industrial applicability of systems that aims to convert the micromanipulation technique to the mass manufacturing process. In order to achieve this goal, the automatic microrobotic system, as the core in the process chain, plays a significant role.

This thesis focuses on the calibration procedure of the positioning control, which is one of the fundamental issues during the automatic microrobotic system development. A novel vision based procedure for three dimensional (3D) calibrations of micromanipulators is proposed. Two major issues in the proposed calibration approach - vision system calibration and manipulator kinematic calibration - are investigated in details in this thesis. For the stereo vision measurement system, the calibration principle and algorithm are presented. Additionally, the manipulator kinematic calibration is carried out in four steps: kinematic modeling, data acquisition, parameter estimation, and compensation implementation. The procedures are presented with two typical models: the matrix model and the polynomial model. Finally, verification and evaluation experiments are conducted on the microrobotic fiber characterization platform in the Micro- and Nano Systems Research Group (MST) at Tampere University of Technology.

The results demonstrate that the proposed calibration models are able to reduce the prediction error below 2.59 micrometers. With those models, the pose error, compensated by the feed-forward compensator, can be reduced to be smaller than 5 µm. The proposed approach also demonstrates the feasibility in calibrating the decoupled motions, by reducing the undesired movement from 28 µm to 8 µm (For 4800 µm desired movement).

PREFACE

This thesis work was carried out in the Micro- and Nano System Research Group (MST) at Tampere University of Technology during December. 2011- May.2012. It was here that I first time got in touch and then fell in love with this amazing field of microrobotics and microengineering. First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Pasi Kallio for his support, guidance and encouragement. Thanks for him to provide such a youthful, flexible and excellent research environment.

Furthermore, it is my great pleasure and honor to study and work in our MST group. My special thanks go to Mathias von Essen and Juha Hirvonen for their insightful suggestions, enlightening help and the fruitful collaboration on my thesis work. I am also grateful to my colleague: Pooya, Yuli and Antti in our microrobotics team.

The thesis is finically supported by the SMARTFIBER project and the FIBAM project from the Finnish Funding Agency for Technology and Innovation -Tekes and the Academy of Finland, which are grateful acknowledged.

Then, I wish to thanks all my Chinese and Finnish friends, epically those I made in Finland, for being always there to support me, and for the unforgettable memory we shared. Thanks for the diligent and friendly Finnish people, the time in this wonderland will be one of the most rewarding treasures in my life.

Last but not least, I am deeply indebted to my dear parents Xiong Zheming and Wang Jinglian for their love over the years.

I dedicate the body of this thesis to the memory of my beloved grandfather, Xiong Gan (1927-2011), for having taught me the value of education and the importance of being grateful. His spirits will with and guide me in the future academic road, forever!

Tampere, May 8th, 2012

Xiong Yi

CONTENTS

LIST OF ABBREVIATIONS

AFM	Atomic Force Microscopy
ANSI	American National Standards Institute
AR	Augmented Reality
ASE	Automation Science and Engineering
CAD	Computer Aided Design
CCD	Charge-Coupled Device
D-H model	Denavit-Hartenberg Model
DLT	Direct Linear Transformation
DOF	Degree(s) of Freedom
EAP	Electro Active Polymers
EPFL	Swiss Federal Institute of Technology in Lausanne
ETH	Swiss Federal Institute of Technology in Zurich
FOI	Field of Interest
ISO	International Organization for Standardization
I/O	Input/ Output
LED	Light-Emitting Diode
MEMS	Micro-Electro-Mechanical Systems
MOEMS	Micro-Opto-Electro-Mechanical Systems
MST	Micro- and Nanosystems Technology
Open*CV*	Open Source Computer Vision
SEM	Scanning Electron Microscope
SMA	Shape Memory Alloy
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition
TEKES	Finnish Funding Agency for Technology and Innovation

TUT	Tampere University of Technology
VR	Virtual Reality

LIST OF SYMBOLS

$\mathbf{A}$	Coefficient vector in the polynomial model
$\boldsymbol{a}_{\boldsymbol{i},\ \boldsymbol{i}=\mathbf{1}\ldots\mathbf{10}}$	Elements in the coefficient vector **A**
$\mathbf{B}$	Coefficient vector in the polynomial model
$\boldsymbol{b}_{\boldsymbol{i},\ \boldsymbol{i}=\mathbf{1}\ldots\mathbf{10}}$	Elements in the coefficient vector **B**
$\{B\}$	Base frame
$\{B_1\}$	Base frame of the left micromanipulator
$\{B_2\}$	Base frame of the right micromanipulator
$\{B_3\}$	Base frame of the micropositioner
$\mathbf{C}$	Coefficient vector in the polynomial model
$\boldsymbol{c}_{\boldsymbol{i},\ \boldsymbol{i}=\mathbf{1}\ldots\mathbf{10}}$	Elements in the coefficient vector **C**
$[\mathrm{c}_x, \mathrm{c}_y]$	Location of principle point on the image plane
$\mathbf{D}$	Sub-matrix of matrix **H**
$\{E\}$	End-effector frame
$\{E_1\}$	End-effector frame of the left micromanipulator
$\{E_2\}$	End-effector frame of the right micromanipulator
$\{E_3\}$	End-effector frame of the micropositioner
$\mathbf{e_i}$	Prediction error of the i^{th} test
f	Focal length in the camera model
f_x	Focal length in the horizontal direction
f_y	Focal length in the vertical direction
$\{G\}$	Global frame
$\mathbf{H}$	4×3 matrix for SVD
$\{I\}$	Image frame
$\{I_1\}$	Image frame of the top camera

$\{I_2\}$	Image frame of the side camera
J	Cost function of the prediction error
$\{J\}$	Jaw frame
$\mathbf{K}$	Intrinsic camera matrix
k_1,k_2	Distortion coefficient for the radial distortion
L	Length of object
l	Length of object on image plane
N	The number of inner corners on chessboard
$\mathbf{P}$	3×4 camera matrix
$\boldsymbol{p}$	Coefficient vector of elements in camera matrix **P**
p_1,p_2	The distortion coefficient for the tangential distortion
$^{I1}\mathbf{P}$	Coordinates in the image frame of top camera $\{I_1\}$
$^{I2}\mathbf{P}$	Coordinates in the image frame of top camera $\{I_2\}$
P_v^X , P_v^Y	Coordinates of the virtual point for x and y axis, respectively
$^{G}\mathbf{P}$	Position vector in the global frame $\{G\}$
$^{G}\mathbf{P}_B$	Position vector of the base frame $\{B\}$ in the global frame $\{G\}$
$^{G}\mathbf{P}_E$	Position vector of the end-effector frame $\{E\}$ in the global frame $\{G\}$
$\{O\}$	Object frame
$\mathbf{R}$	3×3 rotation matrix in the camera model
$\mathbf{T}$	3×1 translation matrix in the camera model
$[\mathrm{t}_1, \mathrm{t}_2]$	Time domain of integration
$\mathbf{U}$	Sub-matrix of matrix **H**
$[u, v]$	Coordinates in the position vector $\boldsymbol{x}_\mathbf{i}$
$[u_i, v_i]$	Coordinates in the image frame of the ith test
$[\hat{u}, \hat{v}]$	Predicted coordinates in the image frame $\{I\}$

$\mathbf{V}$	Sub-matrix of matrix **H**
$\{V\}$	Vision system frame
$[X, Y, Z]$	Coordinates in the position vector $\mathbf{X_i}$
$[x_{d,} y_{d,} z_d]$	Desired coordinates of the end-effector
$[{}^GX_B, {}^GY_B, {}^GZ_B]^T$	Coordinate in the position vector ${}^G\mathbf{P}_B$
$[{}^GX_E, {}^GY_E, {}^GZ_E]^T$	Coordinate in the position vector ${}^G\mathbf{P}_E$
$[X_i, Y_i, Z_i]$	Coordinates in the global frame of the i^{th} test
$\boldsymbol{x}_\mathbf{i}$	Position vector in the image frame $\{I\}$
$\mathbf{X_i}$	Position vector in the global frame $\{G\}$
Z	Distance from the camera to the object
$\boldsymbol{\Theta}$	Joint (actuator) variables vector
λ	Coefficient vector in the parametric kinematic model
Ω	Integration index of undesired displacement

1. Introduction

1.1 Background and Motivation

In the last decade, the development of automatic microrobotic systems has become a thrust area for researchers, and has evolved to a significant sector in microengineering. Microrobotics is a multidisciplinary field that develops miniaturized robotic systems with either overall size or precision in the microscopic level. Currently, the research focuses on two main areas: microrobotic agents and micromanipulation. Micromanipulation concerns about the handling of micro objects ranging from one micrometer to few millimeters. The micromanipulation and microhandling techniques have been successfully applied for a wide range of applications, including:

- living cell manipulation in bioengineering [1] [2],
- micro-electro-mechanical systems (MEMS)/ micro-opto-electro-mechanical systems (MOEMS) assembly in semiconductor manufacturing [3] [4],
- and micro-/nanoscale material characterization in material science [5].

Conventionally, those tasks are conducted manually by skilled operators. However, the process is tedious, time-consuming and with a low success rate. With such a background, it is understandable why high-yield automatic micromanipulation systems have been extensively studied and well-funded in the past ten years. Nowadays, the research on micromanipulation has shifted the scope from the conceptual system development to the industrial applications. Consequently, the future development of this field lies on the industrial applicability of systems that aim to convert the micromanipulation technique to the mass manufacturing process.

Wood derived fibers are cellulosic elements and are used as key raw materials for the paper and pulp industry. Hence, understanding the mechanical properties is of great significance to improve the quality and added value of paper products. However, the typical dimensions of wood derived fibers, ranging from 0.8-4.5mm in length and 16-70 µm in diameter, poses challenges for accurately manipulating those fibers with conventional methods and instruments. Therefore, an automatic micromanipulation system is expected to act as a novel precision instrument for the functionalization, treatment and characterization of individual fibers.

For this purpose, since 2009, a research for the autonomous microrobotic fiber characterization platform development [5] [6] has been launched in the Micro- and Nano System Research Group (MST), Department of Automation Science and

Engineering (ASE), Faculty of Automation, Mechanical and Materials Engineering at Tampere University of Technology (TUT). This initiative research is financially supported by the SMARTFIBER project and the FIBAM project from the Finnish Funding Agency for Technology and Innovation (Tekes) and the Academy of Finland, respectively. By far, the teleoperated fiber characterization process has been successfully achieved and demonstrated on this platform. The next phase of the system development aims to move the system from the teleoperated control level towards the autonomous control level.

Many challenges are encountered while accomplishing the autonomous microrobotic fiber characterization platform. Since the tolerance bands are narrow in microscale, typically under 10 µm for positioning and hundreds µN for force, position and force control are considered as two major challenges. The motivation of thesis work stems from the investigation on improving the positioning accuracy of the microrobotic system. Since the positioning accuracy directly affects whether the devices or target objects can be positioned into the narrow tolerance band, it decides the success or failure of a micromanipulation. In order to improve the positioning accuracy, actions can be taken from either hardware aspect or software aspect. The thesis focuses on using the software-based approach to improve the positioning accuracy.

According to the opinions in [7], calibration and visual servoing are two main streams in the software based approach. Elegant methods to calibrate the position and orientation of a micromanipulator have been developed in [8], [9]. Meanwhile, several other two dimensional (2D) calibration approaches [10] [11] are proposed for the vision guided micromanipulation system in the application area of cell microinjection and two-and-a-half dimensional (2.5D) microassembly. Moreover, a 3D calibration method utilizing a laser sensor is described in [12]. However, such a calibration method which needs to rotate the entire platform is not as suitable in practical operations, since the platform needs to be calibrated whenever the configurations are changed.

Moreover, due to the irregular nature shapes and various dimensions, wood derived fibers and bonds are placed randomly on the fiber bank. Therefore, the 3D locations of their ends are unpredictable and need to be determined during the processes which increase the difficulty of fiber micromanipulation, compared with 1-2.5D micromanipulation tasks.

Motivated by aforementioned reasons, this thesis seeks to develop a generic approach for automatic calibration of microrobotics system based on the 3D vision based measurement. The proposed approach is required to be able to calibrate the micromanipulation system in a relatively short duration within an easy to implement way. After the compensation of all the sources of uncertainty considered, the whole micromanipulation system shall meet the requirements of position accuracy during the usage.

1.2 Contribution

The original contributions of the thesis work are listed as follow:

- The thesis proposes a generic vision based procedure for the automatic calibration of a microrobotic system to improve the 3D position accuracy of the micromanipulation system using a software-based approach.
- The development of a stereo-vision measurement system for the 3D position measurement of micromanipulator end-effector.
- The development of open-loop feedforward control strategies based on the inverse kinematics that derived from the calibrated kinematic model. With this control scheme, the pose error of the micromanipulator can be reduced to be smaller than 5µm.

1.3 Thesis Outline

This thesis discusses the development of a generic vision based approach for the microrobotic system calibration. The rest of this thesis is organized as follows:

In Chapter 2, the background and development in microrobotics research are briefly introduced and discussed. The methods to improve the position accuracy of a microrobotic system from both hardware and software aspects are presented based on the previous classification about sources of uncertainties. The chapter ends with a detail illustration about the frame assignment of interest coordinates in the microrobotic fiber characterization platform.

In order to calibrate a microrobotic system using a vision based approach, the 3D position of the end-effector must be known. Chapter 3 present the development of a vision based 3D pose measurement system from three aspects: Firstly, the parametric model of single camera and the corresponding parameter estimation procedure are described. Then the principles of two-view geometry and the algorithms for 3D reconstruction of calibrated cameras are presented. Since the accuracy of 3D position measurement is of great significance in the following calibration, the accuracy assessment is then performed.

Chapter 4 presents the major contribution of this thesis: a generic vision based procedure for the automatic calibration of the microrobotic system. Following the calibration procedure of industrial robots [13], the approach is illustrated with a four steps procedure: 1) Kinematic modeling; 2) Data acquisition; 3) Parameter identification; and 4) Compensation implementation.

Later, the calibration procedure proposed in Chapter 4 is implemented in an open-loop feedforward control scheme on the microrobotic fiber characterization platform in Chapter 5. Moreover, the decoupling test and pose accuracy are conducted and comparisons are given between the performance results of micromanipulator with and without calibration.

Chapter 6 gives a brief summary of the results and contributions arising from the thesis work. Finally, suggestions to the future work in the field of microrobotics calibration are outlined.

2. Micromanipulation

Depending on the overall size, system precision and task space being macro or micro-scale, the robotic systems can be categorized as macrorobotics and microrobotics, respectively. In the microrobotics, the research mainly focuses on two main subjects: the microrobotic agent and the micromanipulation system. The first area mainly studies the miniaturization issues for robotic agents with the overall size down to sub-millimeter. On the other hand, the investigation in micromanipulation aims to develop autonomous micromanipulation systems which can dexterously handle micro objects, ranging from one micrometer to few millimeters. One should note that, differently for microrobotic agents, the overall sizes of micromanipulation systems are not and unnecessary in the microscopic scale. Typically, the system size is in the meso range, whereas functional parts, system precision and work spaces belong to the micro domain [14].

This chapter includes necessary theoretical background of micromanipulation system. Section 2.1 introduces the background and typical configuration of the micromanipulation system. Section 2.2 discusses six challenges in micromanipulation. Section 2.3 encompasses the position control strategies from both hardware and software aspects. In the software-based approach, three schemes, vision-based feedforward control, vision-based feedback control and augmented reality (AR) control, are introduced and compared. In Section 2.4, the fiber micromanipulation platform studied in this thesis is presented following with the frame assignment of interest coordinates.

2.1 Micromanipulation System

During recent years, micromanipulation, as an emerging and interdisplinary field, is studied by researchers, globally and intensively [15] [16] [17]. In Japan, where is the origin of the micromanipulation concept, a chopstick-like micromanipulator based on a parallel link configuration has been investigated in Prof. T. Arai group [18]. In addition, wide-ranging researches on micromanipulation are also being done at University of Nagoya, Osaka and Tokyo. Research groups in the United States and Canada also make great contributions in this field. Multi-scale assembly systems for MEMS packaging and related precision evaluation have been schematically studied in the group of Prof. Dan O. Popa at University of Texas. Moreover, fruitful results in the application of micromanipulation on biomaterial characterizations have been gained by the group of Prof. Yu Sun at University of Toronto. In Switzerland, intensive researches of micromanipulation have been carried out by the groups of Prof. B.J. Nelson at Swiss federal institute of technology in Zurich (ETH) and Prof. R.Clavel at Swiss federal institute of technology in Lausanne (EPFL) from different levels. Active studies are also being hold in Germany and France. Automatically micro- and nanorobotic systems within tailored scanning electron microscopies (SEMs) or/and atomic force microscopies (AFMs) have been investigated in Oldenburg University. As shown in Figure 2-1, a CAD model based visual tracking approach has been demonstrated by FEMTO-ST for visually guided microrobotics manipulation [19]. In addition, industrial process oriented investigations are being conducted in Finland. In Tampere University of Technology, the unique topic about micromanipulation on bio-fibrous (includes paper fibers and cardiac muscle fibers) are being pursued.

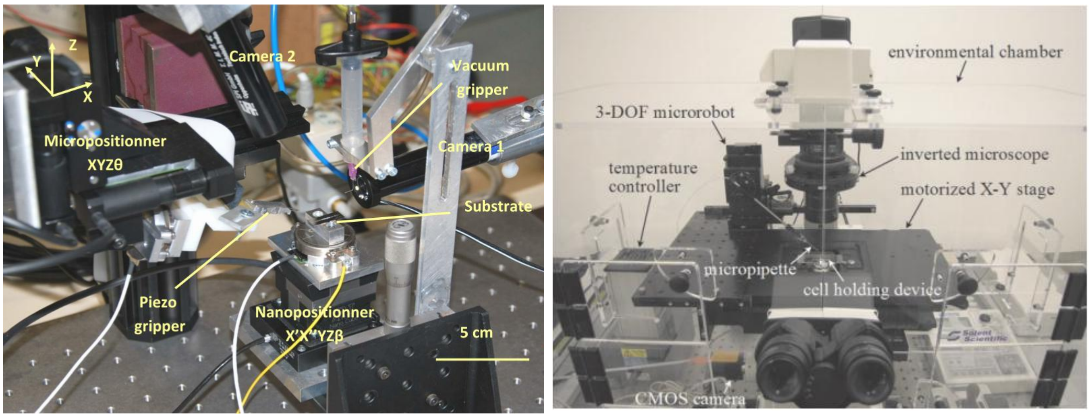


Figure 2-2.1 CAD model based visual controlled microassembly cell with 9 DOF (FEMTO-ST, France) [19]; Micromanipulation system for mouse embryo injection (University of Toronto, Canada) [2].

As mentioned above, several automatic and semi-automatic micromanipulation systems have been reported for various applications. However, a similarity can be found among those systems either in the system configuration or in the control hierarchical architecture.

Generally speaking, a micromanipulation system consists of following sub-systems: one or several micromanipulators, a micro-positioning system with sub-micrometer accuracy, local and global sensing systems, a control unit and other accessories. Since directly performing the required task with micro objects, micromanipulators act as the core in the whole system. From a topological point of view, the reported micromanipulators can be divided into two basic categories: parallel-type micromanipulator and serial-type micromanipulator.

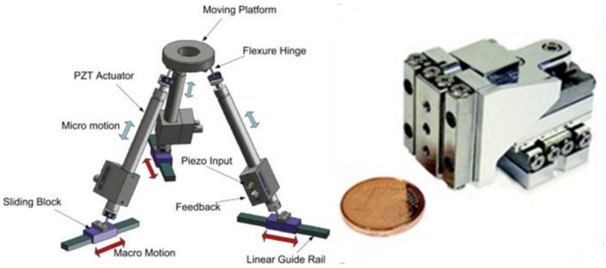


Figure 2.2 3-PUPU parallel micromanipulator (University of Macau, China) [20]; 3DOF Serial micromanipulator (SmarAct GmbH, Germany) [21].

Parallel-type micromanipulators refers to a specific group of micromanipulators that have a closed-chain geometry. The base and end-effector are connected through multiple kinematic chains. In the past five decades, the parallel mechanism has been intensively studied due to the advantages such as unprecedented dexterity, rigidity and higher precision. Thus, in the early stage of micromanipulator development, some well-known parallel mechanism such as Stewart platform and Delta robots have been directly applied to the micromanipulator design. Lately, in order to further remove the joint friction and backlash, the rigid joints in the conventional parallel mechanism are replaced by the flexure hinges in the latest parallel micromanipulator. The major drawback of parallel micromanipulator is the limited workspace. Another drawback is that the complex kinematics of parallel mechanisms increases the difficulty for design.

Other micromanipulators, having only one sequence of links connecting the base and end-effector, are termed as a serial-type micromanipulator. Due to the relative large work space and simplicity in the structure, serial micromanipulators are by far the most common robots. From the rigid body motion theory, the serial micromanipulator with 6 DOF is able to move the end-effector to arbitrary pose in its workspace. However, for the most micromanipulation applications including cell injection and microassembly, 2 or 3 dimensional movements such as move-and-place are already enough. The Gantry Crane and Cartesian configurations, two typical serial-type configurations, are usually adopted in the micromanipulator design. In the microrobotic fiber characterization platform at TUT, all the micromanipulators belong to the serial-type micromanipulator. Hence, the proposed calibration approach in this thesis is mainly for the serial-type micromanipulators.

Furthermore, in order to achieve both actuator control and high-level automation, a three levels hierarchical architecture has been proposed, for example in [12] [22]. As shown in Figure 2.3, the software architecture consists of three layers: the actuator control layer, the planning layer and the mission layer.

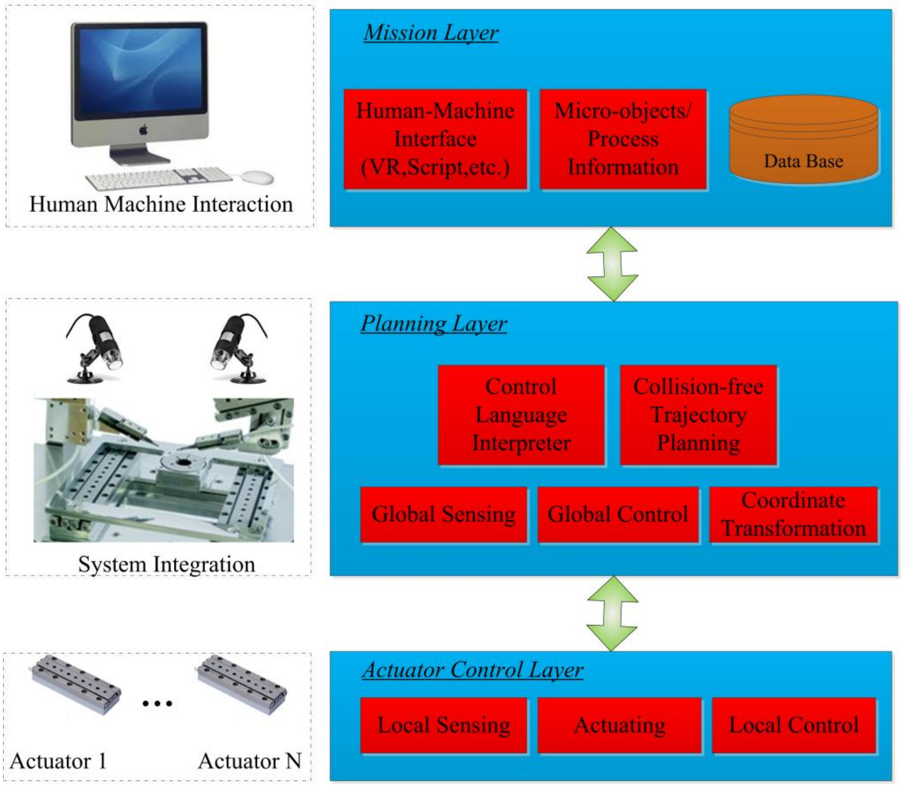

Human Machine Interaction
Mission Layer
Human-Machine Interface (VR,Script,etc.)
Micro-objects/ Process Information
Data Base
System Integration
Planning Layer
Control Language Interpreter
Collision-free Trajectory Planning
Global Sensing
Global Control
Coordinate Transformation
Actuator 1
Actuator N
Actuator Control Layer
Local Sensing
Actuating
Local Control

Figure 2.3 The three levels hierarchical architecture for micromanipulation platform

The actuator control layer is the lowest and basic layer of the whole control system. Because this layer only deals with the individual actuators, it is also called *local control layer* .The main function of this layer is to automatically control the microactuators move to desired positions which are decomposed from up layer. Thus, the performance of this layer has significant influences on the whole accuracy of micromanipulation system. Various control strategies have been used in this layer for actuator control, including pre-shaped open loop control [23], optimal control [24] , and nonlinear inversion control [25] etc. Originally, actuators are typically driven directly in an open-loop fashion by applying input control signals which are determined by the compensator based on the system dynamic. However, with the increasing demands of positioning accuracy, the feedback control with several types of sensors has been introduced. Table 2-1 lists the common used sensor types in the microactuator control.

Table 2-1 Sensor types used in the microactuator control [26]

Sensor Types	***Strengths***	***Drawbacks***
Optical sensors and interferometers	Resolution, accuracy and bandwidth	Expensive, bulky, 1-DOF and pinpoint measurement
Strain Gauges	Reduced sizes, cheap,2-DOF	Fragile, noisy, long time for calibration
Capacitive Sensors	High sensitivity, high Precision, compact sizes	Nonlinear
Piezoelectric Sensors	High band pass, self-sensing possible, embeddable	Nonlinear, difficult for static measurements
Piezomagnetic Sensors		Nonlinear,1-DOF measurement

The planning layer is the mid-layer which interprets the control commands from mission layer and then transforms those into desired set-points in the specified coordinate system. This layer only concerns about the correlations between each actuator, and not about the individual actuator control. Therefore, it is also named as *global control layer*. Different from the local sensing system in the actuator layer, the planning layer usually adopts a microscope with a CCD camera to compose a global sensing system which can provide wide range of information in the micro scale.

One main task of the planning layer is to coordinate the actuators to carry out task in a trajectory without collision. A common practice is to set up a unified world coordinate and then calibrate transformations mapping between the world coordinate and coordinates of each actuator.

The mission layer is located in the uppermost layer which directly with interacts with operator. I/O interface, as the core of this layer, is normally able to provide either script-based commands (e.g. *move micro object A above the micro object B*) or immersion-based commands (e.g. moving the joystick). Moreover, the virtual reality (VR) or augmented reality (AR) system based on CAD model or physical model are developed to provide more accurate perception for the operator.

Since the commercial actuators used in our micromanipulation system already have a local control loop, the thesis will mainly focus on the planning layer. The thesis aims at proposing a generic procedure to calibrate the kinematic model of serial-type micromanipulators. This kinematic model could be used as the basis of the planning layer control later on.

2.2 Challenges in Micromanipulation

With the scaling down size, there are many unique problems that have been raised in micromanipulation. Those challenges make the improving of manipulation accuracy difficult. Generally, the uncertainty comes from six aspects:

- **System Nonlinearity**; Generally, most components in a micromanipulation system such as micro actuators, and micro sensors are based on emerging materials (e.g. Piezoelectric material, Shape Memory Alloy (SMA), Electro Active Polymers (EAP)) .Unfortunately, those material are strongly affected by nonlinearities. Those nonlinearities, including hysteresis and creep, highly affect the positing repeatability and accuracy of the micromanipulation system.

- **Scaling Effect**; When the dimensions scale down, one inherent problem during the micromanipulation with microcomponents comes from the changed physical behavior. In the micro domain, due to the scaling effect, microforces, such as van der Waals, electrostatic and capillary become dominant instead of gravity and inertia [27]. Moreover, due the complex mechanisms of those microforces, the principles and interactions are still not fully understood. Thus, currently, it is even impossible to predict the behavior of microforces with microcomponents.

- **Spatial Uncertainty;** Due to the extremely tiny size with microcomponents, even small disturbances can caused large spatial uncertainty. One cause for the spatial uncertainty in micromanipulation is the thermal drift between the tip and samples [28]. Another important reason for this uncertainty comes from the vibration caused by human, actuators and earth during the operation.

- **Perception;** Restricted by the limited workspace, the interactions between operators (in macro world) and manipulation systems (in micro world) are normally through teleoperation or semi-automation. In a typical micromanipulation system, due to the relatively small space for sensor and controller installation, the information from micro scale is typically recorded through the microscope, transferred to image data and then displayed on PC, after that human or machine will make further decisions based on those information. However, it is quite obvious that the error will propagate through this sensing process which becomes crucial for micromanipulation.

- **Low Signal to Noise Ratio;** In the micro/nano scale, the Signal-to-Noise ratio (SNR) becomes very small due to the very low amplitudes of the signal (typical displacements: 1nm to 100 µm and typical forces: 1 nN to 10 mN) [2]. In some specified displacement range (e.g. less than 100 µm), the magnitude of motion and noise are even in the same order. Therefore, the measurement accuracy, which may high enough in the macro-scale application, is still not sufficient considering the unwanted noise in micro-scale.

- **Ambient Effect;** Based on the previous studies [27], the uncertainties are highly influenced by the ambient environment, such as temperature, relative humidity, mechanical vibration, electromagnetic fields, air cleanliness, air pressure, airflow velocity, illumination condition, etc. However, on one hand it is quite important to regulate the environment to ensure high accuracy manipulation; on the other hand if the ambient effect can be utilized in a proper way, those effects may also have positive effects on the micromanipulation.

2.3 Positioning Control Strategies

In order to achieve accurate and reliable operation with micro-scale objects in the workspace, the micromanipulators need to have a very high positioning accuracy. That is to say, these micro objects should be consistently and precisely located in the same coordinate system. Uncertainties in either end-effector positions or micro objects positions will lead to mismatch. Positioning control aims to reduce those uncertainties by using advanced measurements, signal processing and control strategies. Several methods have been demonstrated, and basically, they can be divided into two categories: hardware-based approach and software-based approach.

2.3.1 Hardware-based approach

From the hardware aspect, one solution is to increase the precision of an actuator by novel actuation principles and mechanisms. In micromanipulation systems, micropositing devices are normally based on the actuation materials such as Piezo ceramics (e.g. Lead Zirconate Titanate), Shape Memory Alloys (e.g. Ti-Ni) and Electro thermal materials (e.g. Diamond like Carbon, SiC, and Allium) which have limited displacements or forces with high nonlinearity [29]. In order to magnify or condition the motions or forces from actuation materials, different actuation principles and compliant mechanisms, including stick-slip drive, inch-worm drive and comb drive have been proposed.

Furthermore, it is also possible to improve the positioning accuracy of the micromanipulator by controlling the ambient variables, including humidity, temperature, and vibrations. In [30], the characterization, modeling and reduction with the effect of environmental noise in a micromanipulation system have been investigated. In another study [31], a microassembly station with a controlled environment and vibration isolation demonstrates a better performance.

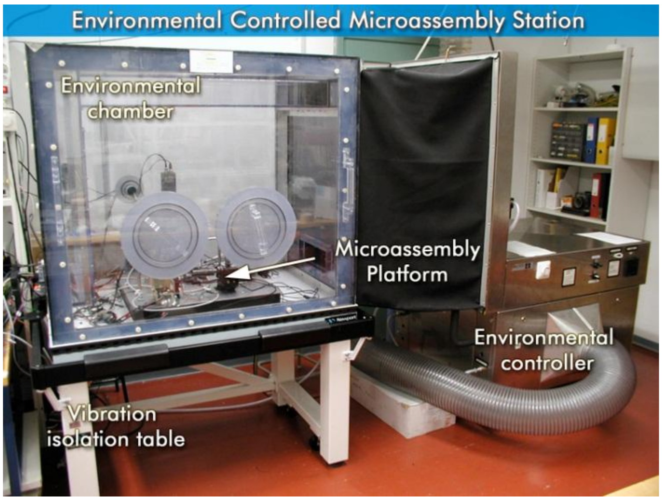

Environmental Controlled Microassembly Station
Environmental chamber
Microassembly Platform
Environmental controller
Vibration isolation table

Figure2.4 Environment controlled microassembly station [31]

There is no doubt to increase the positioning accuracy through the hardware-based approach as much as possible. However, the reduction in system flexibility and productivity are needed to be considered. Nevertheless, the manufacturing imperfectness, alignment errors, and component wearing will inevitably introduce positioning errors from hardware. In addition, the cost is much higher to compensate such errors by a hardware-based approach rather than using a software-based approach.

2.3.2 Software-based approach

Another solution, improving the positioning accuracy, is using the software based approach rather than changing the mechanical structure or design of the robot itself. In last few decades, the vision based control of micromanipulator positioning has been facilitated by the advancement in vision instruments and computer vision technology. General overviews of the main issues and active researches in vision based control of manipulator can be found in [32] [33] [34]. Due to the advantages such as rich information and minimal impact to measurement object, vision based control has also made its own way in the development of micromanipulation systems. A typical vision based control system in the micromanipulation system includes: image acquisition units, signal processing units and a control unit. The image acquisition unit is usually composed of a modified optical microscope with a CCD camera.

According to the control structures, the position of the micromanipulator can be controlled using one of the following control schemes: vision-based feedforward control, vision-based feedback control and augmented reality (AR) control. The first scheme belongs to the open-loop scheme and the others belong to closed-loop scheme.

Figure 2.5 depicts a block diagram of the vision-based feedforward control scheme. In this open-loop fashion, the control scheme consists of an inverse kinematic model and several local control loops for individual actuators. Firstly, the inverse kinematic model is identified by a group of data including both joint inputs and end-effector positions which are measured by machine vision system. Then, the temporal evolution of each actuator inputs is determined by the inversed kinematic model based on the desired location. Therefore, in order to achieve a high accuracy, this kinematic model should be a faithful mathematic represent for the micromanipulator kinematic. In addition, this model should takes account both geometrical and non-geometrical factors which may affect the kinematic relationship. The thesis will concentrate on the derivation of such kinematic model which is also termed as kinematic calibration.

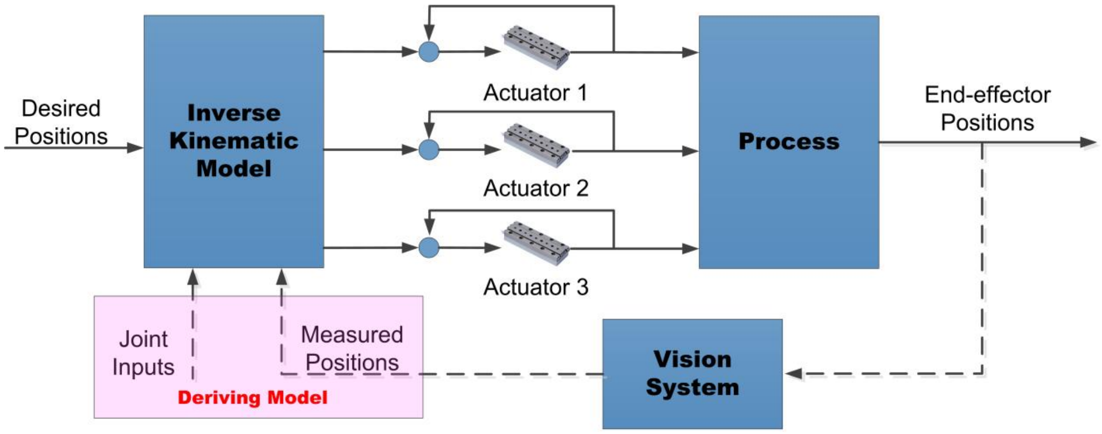


Figure 2.5 Block diagram of vision-based feedforward control

In the vision-based feedforward control scheme, the system static errors such as the manufacturing imperfectness, alignment errors, and component wearing can be partly compensated through the kinematic calibration. However, the unpredictable disturbances and dynamic errors cannot be efficiently eliminated without real-time closed-loop control scheme. Vision-based feedback control, as called visual servoing, refers to the use of visual data within a feedback loop in order to control a manipulating device [35]. As shown in Figure 2.6, similarly to vision feedforward control, an inverse kinematic model identified by visual data is also included in the control loop. However, differently with vision feedforward control, the visual measurement system is still utilized during the closed-loop control process: the feature points of both object and end-effector are first mapped onto the image plane and then measurements of those points are used as the input to the controller. Since the visual system is integrated in the closed control loop, the micromanipulator could be on-line controlled.

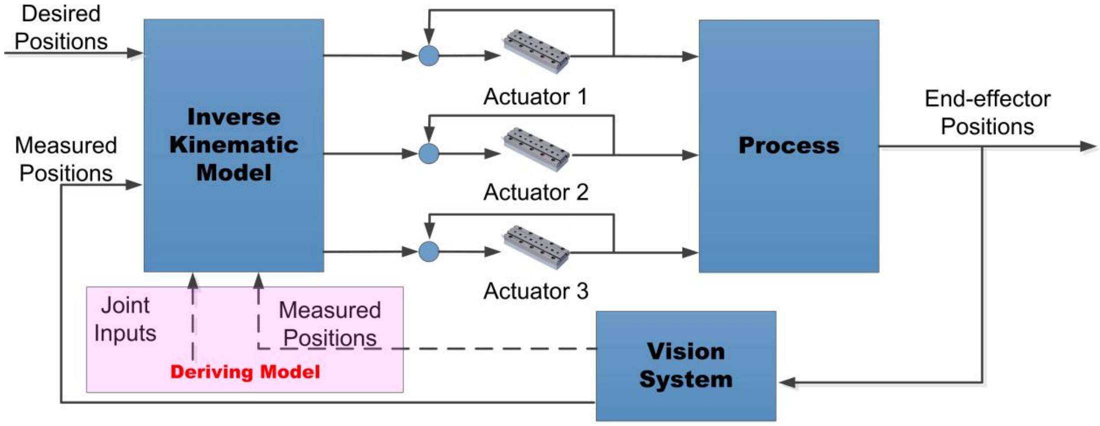


Figure 2.6 Block diagram of vision-based feedback control

Augmented reality is another technique to improve the positioning performance of a micromanipulator. Unlike virtual reality which replace the real environment, the augmented reality supports the real world environment with virtual information. In other words, the augmented reality uses 3D virtual visualization to enhance operator's sense to the physical world. As discussed in Section 2.2, in micromanipulation, the interactions between operator and micro world is limited by the scale and work space. Therefore, the augmented reality is able to solve such limitation.

In all above approaches, the relevant poses between end-effectors and targets are all defined in the image coordinate space. However, through a geometric transformation, this image-based relationship can be also represented in the world coordinate space. Depending on which coordinate space the control signal is referred, there are two possible control strategies: image based-and pose-based.

Images-based servo control uses the location of features on the images directly for feedback. The image-based approach may reduce the computational delay, eliminate the necessity of image interpretation and eliminate errors in sensor modeling and camera calibration. However, on the other hand, the nonlinear and coupled system rises the difficult in the controller design. The other point needed to be considered in this approach is the limited size of the field of interest (FOI) due to the large numerical apertures and high optical magnifications. Normally, the end-effector operation space is larger than the field of interest in the high magnified stationary vision system. In other words, the end-effector may travel out of observation.

In the pose-based approach, the error signal between the actual pose and the desired pose is defined in world coordinate space. One advantage of this approach is that the task can be described in the Cartesian space. Hence, in some complex situations, such as a multi-micromanipulator system in the microassembly process, it is possible to set up a uniform coordinate frames for all the manipulators.

2.4 Microrobotic Fiber Characterization Platform

The platform studied in this thesis is developed as a novel tool for biomaterials characterization (especially for wood-derived fibers and cardiac muscle cells). The objective is to automatically make, manipulate and break fibers and fiber bonds in high throughput.

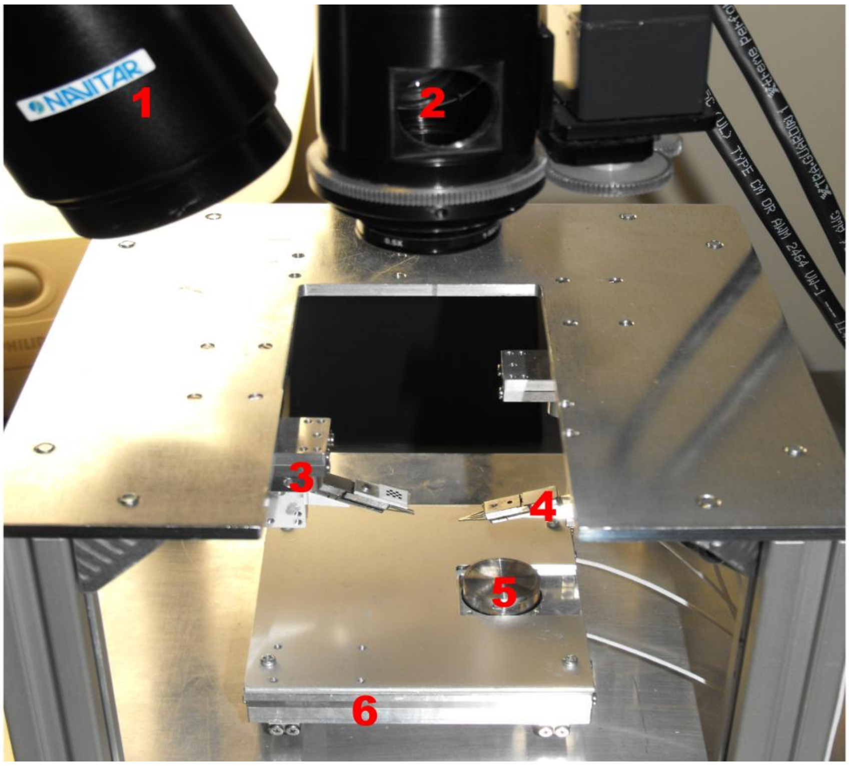


Figure 2.7 The microrobotic platform used for fiber characterization: The top view camera, the side view camera, the micromanipulators (3, 4), the rotary table (5), and the X-Y positioning table (6)

As shown in Figure 2.7, the platform has a Stacked Gantry Crane configuration and includes two tailored SmarAct micromanipulators, an X-Y positioner with a rotary table (SmarAct SR-1908) and a micro-force sensor (Femtotools FT-S54). Each tailored micromanipulator has 4 DOF and is composed of four individual linear actuators. In the same manner, the X-Y positioner with 2 DOF also includes two individual linear actuators. Furthermore, all the linear actuators mentioned above are piezoelectric stick slip type microactuators with a local feedback control system and are purchased from SmarAct GmbH.

In addition, the platform employs two cameras with optics to monitor the operation area from top and side views. The top view camera (XCD-U100 Sony) selected has a 1/1.8 inch CCD cell chip, a 1600 (horizontal) × 1200 (vertical) *pixels* and a pixel size of 4.4 µm. The side view camera (Manta G-504B AVT) employed uses a 2/3 inch CCD cell chip with the resolutions 2452(horizontal) ×2056 (vertical) *pixels* and with a pixel size of 3.45µm. In order to have proper images with high magnification, a motorized zoom system (Magnification ranges from 0.29X to 3.50X) and LED coaxial illumination (Naviator Co.) are also utilized in the system.

In the fiber characterization process, the fibers are initially placed in a fiber bank which is located on the top of the rotary table. The micromanipulators are then used for picking up a fiber from both ends and then synchronously moves to the micro-force sensor for mechanical property characterizations. Since the standard characterization testing requires the relative humidity at 50% ± 2 and the temperature at 20 ± 1°C, the micromanipulation process will be carried out in an environment with controlled humidity and temperature. As discussed in Section 2.2, the positioning accuracy of micromanipulation systems is affected by the ambient environment. Consequently, the derived calibration models in this section are applied for the ambient environment with relative humidity at 50% and temperature at 20°C.

Since the calibration involves several coordinate frames in the micromanipulation system, it is necessary to schematically introduce the definitions and notations first. In this section, coordinate systems are set up and the relations between each system are presented. In addition, the calibration work is also described from the mathematic point of view.

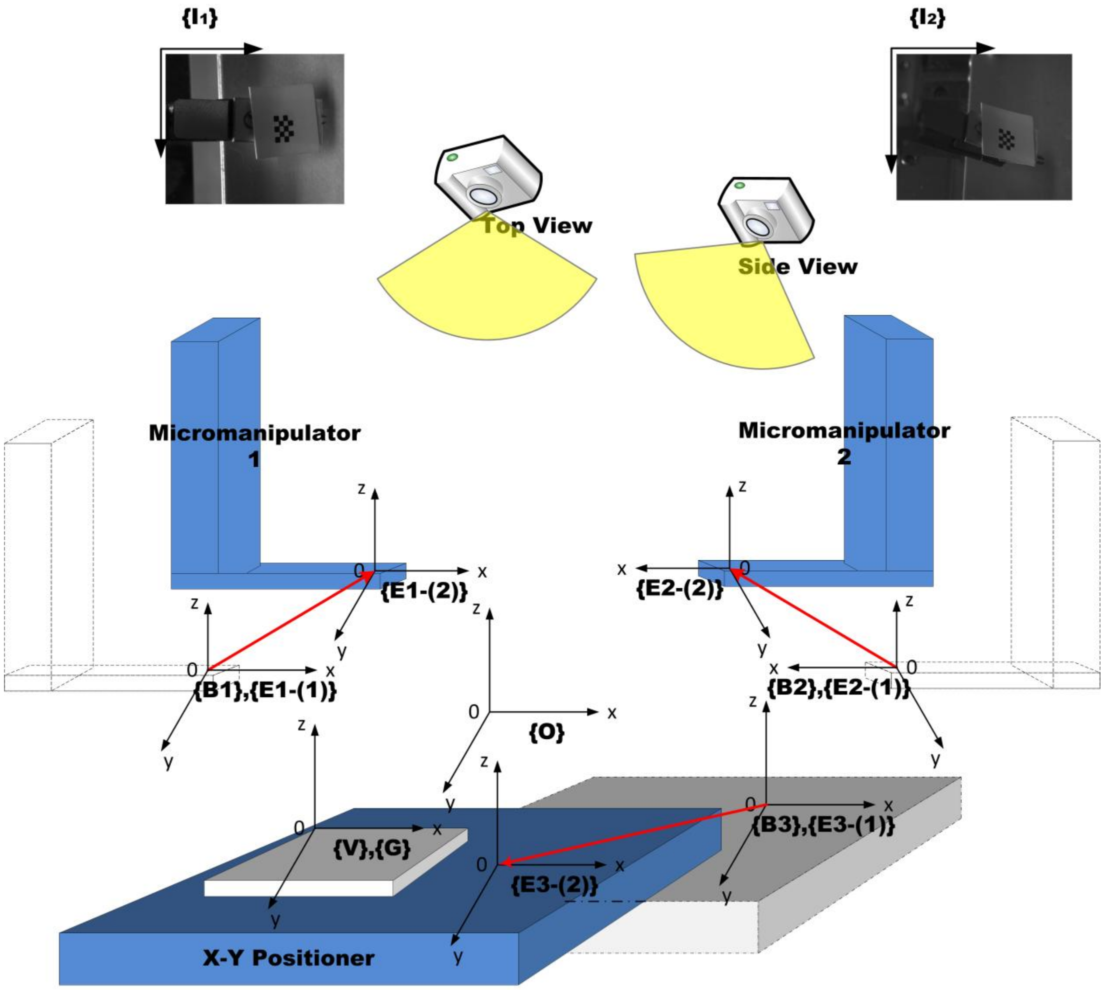


Figure 2.8 Coordinate frames assignment in the microrobotic platform

As shown Figure 2.8, the definition and notations of relevant coordinate system are:

- End-effector frame {*E*}: The origin of the frame is located at a virtual point which is described in Chapter 4 . The *x* and *y* axes are respectively aligned with the rows and columns of a chessboard which is attached to the micromanipulator. {*E1*}, {*E2*} and {*E3*} refer to the end-effectors frame of Micromanipulator 1(No.3 in Figure 2.7), Micromanipulator 2(No.4 in Figure 2.7) and X-Y positioner, respectively.
- Base frame {*B*}: The frame is identical to the initial pose of the end-effector frame {*E-(1)*} when all the joint variables are zero. {*B1*}, {*B2*} and {*B3*} refer to the base frame of Micromanipulator 1, Micromanipulator 2 and X-Y positioner, respectively.
- Image frames {*I1*}, {*I2*}: The image frames describe a 2D coordinate system. They are identical to the local coordinate systems on the CCD image arrays of top and side view cameras, respectively.
- Vision system frame {*V*}: The origin of this frame is affixed to a feature point in a dot grid which is used in the vision system calibration as

described in Section 3. The *x* axis is aligned with the rows of the dot grid; the *y* axis is aligned with the columns of the dot grid.

- Object frame {*O*}: The object frame is located on a desired point of object.

For the control purposes, it is convenient to describe all the points and vectors in a same reference frame. Here, the vision system frame is selected as the global frame {*G*}. The calibration of microrobotics system aims to compute the relationship between the vision system frame and the others, and it comprises two major tasks: vision system calibration and manipulator kinematic calibration.

For vision system calibration, the main task is to reconstruct the 3D coordinate of a feature point in the global frame {*G*} based on the locations of same point from the image frames {*I₁*} and {*I₂*}. In other words, identify the function:

$$^{G}\mathbf{P} = f_1(^{I1}\mathbf{P}, ^{I2}\mathbf{P}) \ , \tag{2.1}$$

where, $^{G}\mathbf{P}$ is the 3D coordinate of an interest point in the global frame. $^{I1}\mathbf{P}$ and $^{I2}\mathbf{P}$ are the corresponding 2D coordinates of the interest point in the image frame {I1} and image frame{*I2*}, respectively.

The manipulator kinematic calibration aims to investigate the transformation from the end-effector frame {*E*} to the base frame {*B*} which described the kinematic of the micromanipulator. That is say, estimate the function:

$$^{G}\mathbf{P}_E = f_2(^{G}\mathbf{P}_B, \mathbf{\Theta}, \boldsymbol{\lambda}), \tag{2.2}$$

where, $^{G}\mathbf{P}_E$ and $^{G}\mathbf{P}_B$ are the coordinate in the end-effector frame {*E*} and the base frame {*B*} in the global frame, respectively. $\mathbf{\Theta}$ is the joint variable vector. In addition, $\boldsymbol{\lambda}$ is the coefficient vector in the parametric kinematic model.

The rest of thesis will be organized based on the derivation of those two transformations. Section 3 presents the stereo-vision system calibration based on Equation 2.1, and Section 4 introduces the manipulator kinematic calibration based on Equation 2.2.

2.5 Summary

Micromanipulation refers to the manipulation of micro-scale objects. As an interdisciplinary field of robotics and microsystem technology, the emerging micromanipulation systems are able to achieve accurate, automatic and effective micromanipulation. However, several issues that limit the accuracy of micromanipulation include: system nonlinearity, scaling effect, spatial uncertainty, limited perception, low signal to noise ratio and effects from ambient. In order to improve the system behavior, actions can be taken from either hardware aspect or software direction. From the hardware aspect, advanced actuator principles, accurately manufactured and assembled components and controlled ambient are needed. In software direction, system calibration, advanced control scheme and signal conditioning. have been widely studied and used. Vision based approach as a software approach, has the advantages of rich information, minimal impact to measurement objects.

Currently, most utilized vision based approach includes vision-based feedforward control, vision-based feedback control and augmented reality (AR) control. Among all three methods, the inverse kinematic model of robotic process plays a significant role. This thesis focuses on the representing and deriving of kinematic model. Two representation forms including matrix model and polynomial model are derived utilizing a stereo vision measurement system.

3. Vision Based 3D-Pose Measurement

As discussed in Chapter 2, due to the limited workspace for sensor installation, non-contact measurement approaches, such as vision-based and laser-based, are widely utilized for sensing in micromanipulation systems. Compared with other non-contact measurement methods, vision based approach provides an accurate enough measurement within a compact system setup. Moreover, off-the-shelf components are available on the market for rapid system prototyping at a relative low cost.

Typically, the vision based measurement system in micromanipulation applications employs a microscope-camera system configuration. Within such system configuration, the image of monitor and/or control objects in the micro-scale are magnified through the objective and tube-lens, and then projected onto the image sensor plane (a CCD array). However, based on the magnified image from one CCD camera, it is only possible to detect the 2D location (x, y) of an object in the 3D world space. In order to recover the lost depth information about z axis in the 3D world space, several well-honed approaches exist:

a) For the single-camera configuration (also called mono-vision), autofocus technique can be employed to extract z coordinate by adjusting the object lens to obtain fine images and the focus position.
b) Utilizing a multi-view setup to obtain the z coordinate by epipolar geometry.
c) Another approach that also employs multi-view is the special case of method b) which configures two cameras in 90º orientation.

In this thesis work, the approach b) has been adopted to reconstruct the 3D information.

Above all, the 3D world coordinate locations of the target objects must be inferred from the magnified images through a transformation. This transformation can be described with a parametric model. In the following section, the parametric geometry model of individual cameras and a corresponding parameter estimation method will be introduced in Section 3.1. Based on two-view geometry, Section 3.2 elucidates the procedure for three dimensional reconstruction. Finally, the measurement accuracy will be evaluated in Section 3.3.

3.1 Camera Geometry

3.1.1 Parametric Model

In the past decades, benefited from the advance in computer vision, several significant literatures that deal with the parametric camera model can be found. In Tsai's classic paper [36], a camera model with extrinsic parameters and intrinsic parameters has been proposed based on the pinhole model.

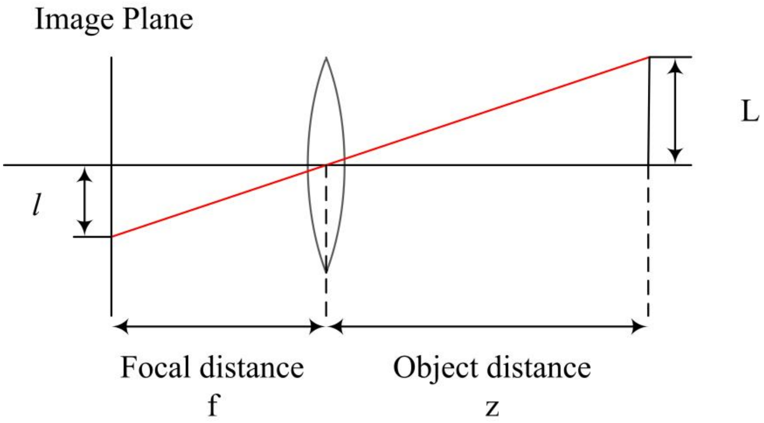


Figure 3.1: Pin-Hole camera model

As illustrated in Figure 3.1, the pinhole model is the simplest representation for a camera image system. In this model, an envisioned single ray from a distant object project to the *image plane* (also called the projective plane) through the focus (a particular point which those light rays intersect).

For an idealized pinhole camera, the distance from the pinhole aperture to the image plane is called *focal length*. Based on the similar triangles, it is easy to get the relationship between the object length L and corresponding length l in the image plane:

$$l = f\frac{L}{Z}, \tag{3.1}$$

where, f is the focal length, Z is the distance from the camera to the object, l and L are the object size and the corresponding image, respectively. The ratio between l and L is called magnification. It is obvious that if we keep the distance from the camera to the object constant, then the larger the focal length f is, the higher the magnification will be.

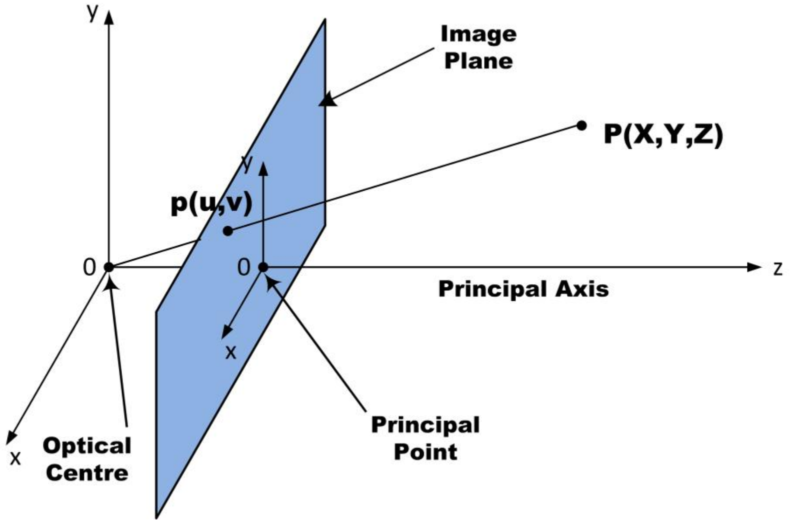


Figure 3.2 Point P (X,Y,Z) in 3D world space is projected onto the image plane at the point p(u, v)

For convenience purposes, the above model is usually modified by pushing the image plane in front of the pinhole at the same distance from it as the actual image plane to avoid inverted images. In order to model the idealized camera, two coordinate systems are set up in this model:

- Objective coordinate system (*X, Y, Z*): The origin of the frame is identical to the optical center. The *X* axis is aligned with the rows of the image frames, the *Y* axis is aligned with the columns of the image frame, and the *Z* axis is aligned with the optical axis.
- Image coordinate system (*u, v*): The origin of the frame is located at the intersection between the virtual image plane and the optical axis. *u* and *v* are chosen parallel with *X* and *Y* in the objective coordinate system.

For a point (*X, Y, Z*) in the objective coordinate system, as depicted in Figure 3.2., the corresponding location of the projected point in the image coordinate system (*u, v*) can be derived through the following transformation:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{3.2}$$

However, in practice, this model needs to be modified from the following aspects to compensate the error in lens manufacturing and installation:

- **Non-square pixels;** In the case of CCD cameras, it has to be considered that the numbers of pixels in horizontal and vertical of digital detector are not exactly the same, so that the scale factors are not equal in two axial directions.

- **Principle points;** From a physical point of view, the reference axis for the camera model is not the same as the optical axis, whereas a principle ray that is perpendicular to the image plane intersects the latter at the principle point [37].
- **Distortion;** Most practical lenses are not as perfect as an ideal lens which render the straight light straightly. Due to the shape of a lens, rays further from the center of the lens are bent more than close to the center (ideally zero at center). This distortion is called the *radial distortion*. Moreover, the manufacturing defects resulting from the lens not being exactly parallel to the image plane will also cause the *tangential distortion*. Typically, the effect of tangential distortion is much less than radial distortion.

In addition, one rule of thumb is that the point is more convenient to be described in the world coordinate frame rather than in objective coordinate system. The transformation between two frames includes a rotation matrix **R** and a translation matrix **T**. Considering the above influence factors, the parametric model for a camera image system can be represented as following:

$$\begin{aligned}
&\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{T} \\
&x' = x/z \\
&y' = y/z \\
&x'' = x'(1+k_1r^2+k_2r^4)+2p_1x'y'+p_2(r^2+2x'^2) \\
&y'' = y'(1+k_1r^2+k_2r^4)+p_1(r^2+2y'^2)+2p_2x'y' \\
&r^2 = x'^2+y'^2 \\
&u = f_x * x'' + c_x \\
&v = f_y * y'' + c_y
\end{aligned} \quad , \tag{3.3}$$

where, the parameters can be divided into two groups:

Extrinsic parameters:

1) **R** is a 3×3 rotation matrix;
2) **T** is a 3×1 translation matrix.

Intrinsic parameters:

1) f_x is the focal length in horizontal direction;
2) f_y is the focal length in vertical direction;
3) [c_x, c_y] is the location of principle point which is the point where the optic axis intersects the image plane;

4) k_1,k_2 are the distortion coefficients for the radial distortion;
5) p_1,p_2 are the distortion coefficients for the tangential distortion.

After the distortion compensation, this model can be simplified as follows:

$$\mathbf{x} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}_{\mathbf{3x1}} & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \qquad (3.4)$$

Now, denoting the intrinsic matrix

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \qquad (3.5)$$

Consequently, the 3D to 2D camera projection has a concise form:

$$\mathbf{x} = \mathbf{PX} \qquad \mathbf{P} = \mathbf{K}\langle \mathbf{R} | \mathbf{T} \rangle = \mathbf{KR}\langle \mathbf{I} | \mathbf{R}^{\mathbf{T}}\mathbf{T} \rangle, \qquad (3.6)$$

where, **X** and **x** are coordinate vectors of the 3D point and the corresponding image point, respectively. **P** represents the 3×4 camera matrix.

3.1.2 Parameter Estimation

Based on Equation 3.6 that represents 3D to 2D camera projection, this section describes the numerical approaches for estimating the camera matrix **P** from a set of correspondences. The main task of parameter estimation can be stated as:

Given:

n correspondences ($\mathbf{x_i}$, $\mathbf{X_i}$), where $\mathbf{X_i}$ is the location of a point in Euclidean 3-space and $\mathbf{x_i}$ is the location of corresponding point in image space.

Compute:

the 3x4 camera matrix $\mathbf{P} = \mathbf{K}\langle \mathbf{R} | \mathbf{T} \rangle$ such that $\mathbf{x_i} = \mathbf{PX_i}$

Since the relationship between $\mathbf{x_i}$ and $\mathbf{X_i}$ is linear, thus the camera matrix **P** which has 12 elements can be solved by 12 linear dependent equations. From one set of correspondences ($\mathbf{x_i}$, $\mathbf{X_i}$), it leads to three equations, and thus at least four sets are needed to compute all the parameters.

Equation 3.6 can be rewritten in terms of the elements in matrix P as follows:

$$\mathbf{x}' = \mathbf{X}'\mathbf{P}, \qquad (3.7)$$

where, $\mathbf{x}' = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}, \mathbf{X}' = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 \end{bmatrix},$

and the 12×1vector

$\boldsymbol{p} = [P_{11} \quad P_{12} \quad P_{13} \quad P_{14} \quad P_{21} \quad P_{22} \quad P_{23} \quad P_{24} \quad P_{31} \quad P_{32} \quad P_{33} \quad P_{34}]^T$ by rearrange the elements in the camera matrix $\mathbf{P}$.

Augmenting by two other correspondences, the whole 12 linear dependent equations can be expressed as:

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \\ u_2 \\ v_2 \\ 1 \\ u_3 \\ v_3 \\ 1 \\ u_4 \\ v_4 \\ 1 \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 \\ X_3 & Y_3 & Z_3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_3 & Y_3 & Z_3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_3 & Y_3 & Z_3 & 1 \\ X_4 & Y_4 & Z_4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_4 & Y_4 & Z_4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_4 & Y_4 & Z_4 & 1 \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ P_{21} \\ P_{22} \\ P_{23} \\ P_{24} \\ P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \end{bmatrix}. \quad (3.8)$$

Since the matrix $\mathbf{X}'$ is a non-singular 12 x 12 square matrix, thus the matrix $\boldsymbol{p}$ can be solved by:

$$p = \mathbf{X}'^{-1}\mathbf{x}', \quad (3.9)$$

The 3×4 camera matrix **P** can be then derived by rearranging the elements of the matrix ***p***. However, since the measurement of the image coordinates contains error, it is impossible to determine a matrix **P** that leads the prediction error $\mathbf{e_i}$ to be zero:

$$\mathbf{e_i} = \mathbf{x_i} - \widehat{\mathbf{x}}_\mathbf{i} = \mathbf{x_i} - \mathbf{PX_i}, \quad (3.10)$$

where, $\mathbf{e_i}$ is the prediction error, $\widehat{\mathbf{x}}_i$ is the predicted coordinate of 3D space point $\mathbf{X_i}$ and $\mathbf{x_i}$ is the measured coordinate of same 3D space point.

Thus, an estimate that minimizes the cost function of the prediction error is computed instead of seeking an exact solution of **P**. Therefore, enough (more than 4 sets) measured data is needed, so that the number of equations is larger than the number of unknowns. The influence of measurement error on the computation results will be reduced to an accepted level if the number of measured data is larger enough. One common scalar cost function of prediction error J is created as:

$$J = e^T e = (\mathbf{x_i} - \mathbf{PX_i})^{\mathbf{T}}(\mathbf{x_i} - \mathbf{PX_i}), \tag{3.11}$$

where, *J* is the cost function, which is also termed as object function. Since the performance criterion *J* is a quadratic function of **X**, the minimum value of *J* is obtained when the partial derivation respect to **X** is zero:

$$\frac{\partial J}{\partial \mathbf{X}} = 0, \tag{3.12}$$

and the Hessian matrix of *J* is positive semi-definite:

$$\frac{\partial^2 J}{\partial^2 \mathbf{X}} \geq 0. \tag{3.13}$$

The solution that meets the above conditions is:

$$\mathbf{X^T XP} = \mathbf{X^T x}. \tag{3.14}$$

If the square matrix $\mathbf{X^T}$ **X** is non-singular, then the equation yields a least-squares estimate as:

$$\mathbf{P} = (\mathbf{X^T X})^{-1}\mathbf{X^T x}, \tag{3.15}$$

where, the form $(\mathbf{X^T X})^{-1}\mathbf{X^T}$ is called the Moore–Penrose pseudo inverse of the matrix **X**. The pseudo inverse is a generative inverse matrix for the non-square matrix.

In addition, the intrinsic parameters **K** and extrinsic parameters **T** and **R** can be computed by decomposing the estimate of the camera matrix **P**. The first 3 × 3 submatrix **M** of **P** is the product (**M=KR**) of an upper triangular and rotation matrix:

1) Factor **M** into **K** and **R** via the *QR* matrix decomposition. This determines **K** and **R**.
2) Then, the translation matrix **T** can be computed by:

$$\mathbf{T} = \mathbf{K}^{-1}(P_{14}, P_{24}, P_{34})^T. \tag{3.16}$$

3.2 Three Dimensional Reconstruction

In the previous section, a perspective camera is represented by a 3 × 4 matrix **P**. The elements of the matrix are computed from the correspondence of 4 (or more) points. However, since the depth information is lost during transformations, the location of a point in Euclidean 3-space cannot be computed only from a single view. Thus, in order to reconstruct the 3D coordinate, two or more views are needed. This thesis will only focus on the 3D reconstruction from two views.

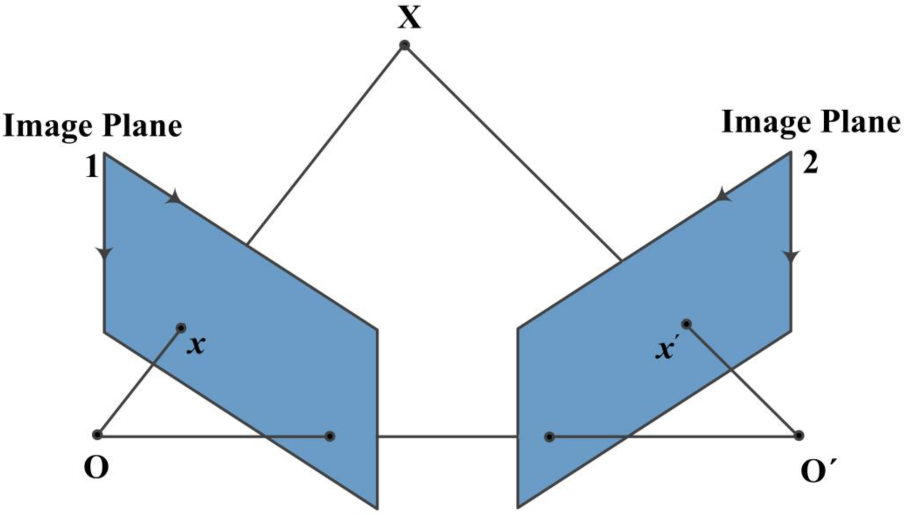


Figure 3.3 The geometry of stereo imaging system

As shown in Figure 3.3, in essence, the stereo imaging system comprises two individual cameras which pose at different locations. The point x and x' are both 2D images of the point X in 3D space. With the previous knowledge in Section 3.1, each camera can be described with a 3×4 camera matrix:

$$\begin{aligned}\mathbf{x} &= \mathbf{P}\mathbf{X} \\ \mathbf{x}' &= \mathbf{P}'\mathbf{X}\end{aligned}. \tag{3.17}$$

Typically, the next step for 3D reconstruction is to compute point correspondences based on the epipolar geometry. However, as a dot grid is here used, it is unnecessary to find out the point correspondences. Thus, this step will not be included in this case.

If the n-th row of the matrix $\mathbf{P}$ ($\mathbf{P}'$) is denoted by p^{NT}(p'^{NT}), Equation 3.17 can be then rewritten as linear equations in X:

$$\mathbf{H}\mathbf{X} = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix} \mathbf{X} = 0\,. \tag{3.18}$$

This equation can be solved by the direct linear transformation (DLT) algorithm to obtain the least-square solution [38]. Therefore, the algorithm for reconstructing the 3D coordinates in the vision frame can be described as:

1) Form equation **HX**=**0**

2) Decompose the matrix **H** into three sub matrices **U**, **D**, $\mathbf{V}^{\mathbf{T}}$ using the Singular value decomposition (SVD).
3) Since **H** has the dimension 4 × 3 and rank 3, and it has a 1-dimensional null-space which provides a solution for **X**.

Consequently, the homogenous coordinates of the 3D point after reconstruction is derived as the smallest singular value of matrix **H** which is the last column of **V**.

In order to evaluate the performance of the derived algorithm, the reprojection error is introduced. As shown in Figure 3.4, the computation of reprojection error is an inverse process of above derivations from 2D images to 3D coordinates. The point X in 3D space is projected to points $\hat{x}$ and $\hat{x}'$ on the Image Plane 1 and 2 through the inverse transformation of matrices P and P'.

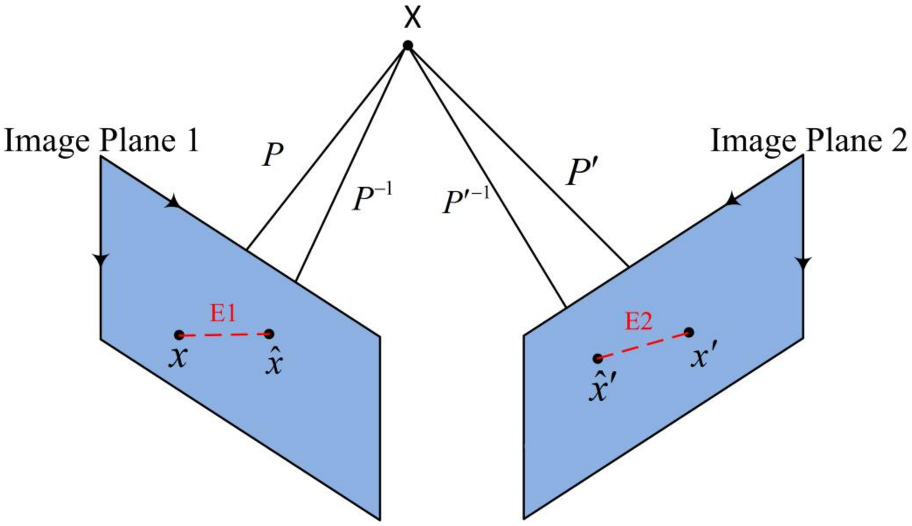


Figure 3.4 The process of 3D point reprojection

The reprojection error is defined as the spatial distance between the measured point x and predicted point $\hat{x}$ and it is computed as:

$$RE = \sqrt{(u-\hat{u})^2 + (v-\hat{v})^2} \ , \tag{3.19}$$

where, (u, v) is the measured point location in the image frame, and $(\hat{u}, \hat{v})$ is the predicted point location in the image frame.

3.3 Measurement Accuracy Assessment

Section 3.1 and Section 3.2 describes the background and algorithms of the stereo vision measurement system. This section discusses the issues related with the accuracy investigation of the aforementioned system.

In order to evaluate the performance of a vision based measurement system, the terms camera sensor resolution, spatial resolution and measurement accuracy need to be distinguished [37]:

- Camera Sensor Resolution: For a CCD camera, this refers to the numbers of detector cells in columns and rows. The unit is pixels. In addition, the detector also affects the resolution.
- Spatial Resolution: This resolution concerns the direct mapping of real-world object to the CCD sensor. The unit is mm/pixel. It can be estimated by taking the camera sensor resolution, and dividing it by the magnification.
- Measurement Accuracy: This is the overall performance of the vision measurement system and it determines the minimum feature that can be measured. Since the measurement system is composed of optics, electronics and software, the measurement accuracy depends on not only hardware resolution such as spatial resolution but also software performances. The Table 3-1 lists out typical feature detection algorithms and their expected accuracy. In the proposed vision measurement system, *cv2.cornerSubPix* function in *Opencv* library is used to refine the corner feature locations. The details of the algorithm can be found in [39].

Table 3-1Typical feature detection algorithm and their expected accuracy

Algorithm	***Accuracy (pixel)***
Edge Detection	1/3
Blob	3
Pattern Matching	1

Since the measurement accuracy is affected by several factors, it is more straightforward and precise to estimate the measurement accuracy from experiments than the analytical derivation. However, since 3D position measrument with nanometer precision is still challenging, it is difficult or even impossible, to accurately evaluate the performance of measurement. Since the reprojection error (unit in pixel), discussed in the previous section, can represent the measurement accuracy from another aspect, errors in the image frame, the exact measurement accuracy(unit in micrometer) is not examined.

3.4 Summary

This chapter presents a micromanipulator pose measurement system based on stereovision and related measurement accuracy assessment. The algorithms of the stereovision measurement system are described in details. Firstly, the camera model and camera matrix estimation are presented. Then, utilizing two calibrated individual cameras, the procedure of 3D reconstruction from two views is described. Finally, the issues related with the measurement accuracy of the stereovision system are shortly discussed.

4. Micromanipulation System Calibration Approach

As discussed in Section 2.4, the vision based micromanipulation system calibration includes two main parts: vision system calibration and manipulator kinematic calibration. In the previous chapter, the vision based measurement system calibration has been presented. Based on the previous calibrated visual measurement system, this chapter focuses on the micromanipulator kinematic calibration.

According to the definition in [40], the kinematic calibration refers to the process that improves the positioning accuracy through a software approach rather than changing or altering the design of the robot. It involves identifying a more accurate functional relationship between the joint variables and the actual position of end-effector and using this identified model to construct the control algorithm for micromanipulator.

In general, the kinematic calibration in either macro or micro scale consists of four steps: The first step would be to choose a suitable mathematic model that represents the functional relationship. This step is termed as kinematic modeling. Next step which is data acquisition collects enough data from the actual joint inputs and positions of the end-effector. The third step called parameter estimation deals with a mathematic process using collected data to estimate the parameters in the model. Finally, the micromanipulator compensation will be implemented based on the identified model. This step is referred as compensation implementation.

Following the typical kinematic calibration procedure, the chapter is divided into five sections. Section 4.1 introduces the kinematic modeling after a brief literature review. Section 4.2 presents the data acquisition through calibrated visual measurement system. Section 4.3 describes two typical estimation algorithms for the matrix model and polynomial model. Section 4.4 explains the compensation implementation for practical purpose. Finally, Section 4.5 is for discussion of all contents in this chapter.

4.1 Kinematic Modeling

4.1.1 Literature Review

Kinematic modeling, as the first step in the calibration process, is the determination of a suitable model which utilizes a mathematical description of the relationship between the motion of robots and related geometry and non-geometry parameters.

Before introducing the details of models, it is necessary to first briefly discuss the properties that a suitable model should have for the kinematic calibration. In [41], the author claims that a suitable kinematic model for calibration should meet three criteria:

- ***Completeness.*** To be complete, the model should has enough coefficients to express any variation of the actual robot structure away from the nominal design caused by either geometry or non-geometry parameters.
- ***Proportionality.*** Proportionality implies that small changes in the robot system should be reflected by small changes in the kinematic model. In other words, the model should be sensitive enough to detect small changes in the system.
- ***Equivalence.*** Equivalence refers to the ability to apply the model from one robot system to another structurally similar robot system with only parameter changes.

Several well-honed approaches to develop models with above requirements have been proposed, including the Denavit-Hartenberg (D-H) model. According to the model structures, the modeling methods can be roughly divided into two groups: *white box modeling* and *black box modeling* [42].

The first approach is mainly based on the direct or inverse kinematic model derived from geometric relationship. This analytical model describes the motion propagation in the robot kinematic chain. Non-geometric parameters are usually introduced to the model by adding extra variation terms to the overall geometric model of the manipulator [2]. The principle of this method determines that this method will explicitly express the kinematic structure.

For the black box modeling, errors in the relative position and orientation of the end effector are measured for a suitable number of points within the working capacity. These measurements can be repeated under various conditions. Using those data, an empirical model is estimated based on the input and output information. However, this model, whose parameters are basically vehicle for adjusting the fit to data, does not reflect any physical relationship. Therefore, this approach is an implicitly method which “mask” the geometry relationship into coefficients of the model.

Since the joint geometric dimensions are difficult or even impossible to be accurately measured in micro scale, geometry based modeling is not suitable for our case. Consequently, in the following section, the black box model is employed for the kinematic modeling.

4.1.2 Matrix Model

The most widely used way to represent the transformation between two frames is using the matrix model as following:

$$^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG}\,, \tag{4.1}$$

where, $^{A}P_{BORG}$ is the 3×1 translation matrix and $^{A}_{B}R$ is the 3×3 rotation matrix. The spatial distance between the origins of two frames is denoted by the translation matrix. On the other hand, the rotation matrix describes the orientation of the coordinate frame {A} relative to the coordinate frame {B}. Additionally, every rotation matrix could be decomposed into three elementary rotation matrixes. The rotation of frame about the x axis through an angle θ is represented as

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}\,, \tag{4.2}$$

while the same rotation about the y axis is

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}\,, \tag{4.3}$$

and about the z axis is

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\,, \tag{4.4}$$

However, as mentioned above, it is impossible to measure the rotation angle between two frames. Therefore, the matrix model, in a black-box form, is selected as:

$$\begin{bmatrix} ^{G}X_E \\ ^{G}Y_E \\ ^{G}Z_E \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} + \begin{bmatrix} ^{G}X_B \\ ^{G}Y_B \\ ^{G}Z_B \end{bmatrix}, \tag{4.5}$$

where, $[^{G}X_E, {}^{G}Y_E, {}^{G}Z_E]^T$ and $[^{G}X_B, {}^{G}Y_B, {}^{G}Z_B]^T$ are the locations of the end-effector frame {E} and base frame{B} in the vision frame {V}, respectively. Vector $\mathbf{\Theta}$ = [d_x,

$d_{y,}d_z]^T$ represents the joint variables of the individual actuators along *x*, *y* and *z* axis. The 9 unknown parameters, which need to be experientially identified, are denoted by matrix $\boldsymbol{R}$=[r$_{ij}$](i,j∈[1,3]).

4.1.3 Polynomial Model

In the matrix model, the relationship between end-effector position and individual joint variables is assumed to be linear. However, in practice, some geometric or non-geometric factors may not only 1st order function but also higher order function of joint variables. Thus, by extending the matrix model, the 2nd order terms of joint variables have been introduced in the model as follows:

$$\begin{bmatrix} {}^G X_E \\ {}^G Y_E \\ {}^G Z_E \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} + \begin{bmatrix} a_4 & a_5 & a_6 \\ b_4 & b_5 & b_6 \\ c_4 & c_5 & c_6 \end{bmatrix} \begin{bmatrix} d_x{}^2 \\ d_y{}^2 \\ d_z{}^2 \end{bmatrix} + \begin{bmatrix} a_7 & a_8 & a_9 \\ b_7 & b_8 & b_9 \\ c_7 & c_8 & c_9 \end{bmatrix} \begin{bmatrix} d_x \cdot d_y \\ d_y \cdot d_z \\ d_x \cdot d_z \end{bmatrix} + \begin{bmatrix} a_{10} \\ b_{10} \\ c_{10} \end{bmatrix} + \begin{bmatrix} {}^G X_B \\ {}^G Y_B \\ {}^G Z_B \end{bmatrix}, \qquad (4.6)$$

where, [^GX_E, ^GY_E, ^GZ_E]^T and [^GX_B, ^GY_B, ^GZ_B]^T are the locations of the end-effector frame {*E*} and base frame{*B*} in the vision frame {*V*}, respectively. Vector $\boldsymbol{\Theta}$ = [d_x, $d_{y,}d_z$]^T represents the joint variables of the individual actuators along *x*, *y* and *z* axis. In addition, 30 unknown parameters are defined with three vectors: **A**= [$a_{1,}$ a_2...a_{10}], **B**= [$b_{1,}$ b_2...b_{10}], and **C**= [$c_{1,}$ c_2...c_{10}].

By expanding Equation 4.6, the coordinates of the end-effector can be determined in the vision frame as follows:

$$\begin{aligned} {}^G X_E &= a_1 d_x + a_2 d_y + a_3 d_z + a_4 d_x{}^2 + a_5 d_y{}^2 + a_6 d_z{}^2 + a_7 d_x d_y + a_8 d_y d_z \\ &\quad + a_9 d_x d_z + a_{10} + {}^G X_B \end{aligned} \qquad (4.7)$$

$$\begin{aligned} {}^G Y_E &= b_1 d_x + b_2 d_y + b_3 d_z + b_4 d_x{}^2 + b_5 d_y{}^2 + b_6 d_z{}^2 + b_7 d_x d_y + b_8 d_y d_z \\ &\quad + b_9 d_x d_z + b_{10} + {}^G Y_B \end{aligned} \qquad (4.8)$$

$$\begin{aligned} {}^G Z_E &= c_1 d_x + c_2 d_y + c_3 d_z + c_4 d_x{}^2 + c_5 d_y{}^2 + c_6 d_z{}^2 + c_7 d_x d_y + c_8 d_y d_z \\ &\quad + c_9 d_x d_z + c_{10} + {}^G Z_B \end{aligned} \qquad (4.9)$$

.

Compared with other models, the polynomial model has several advantages. First of all, this model can be extended by adding extra terms if additional factors show significant impacts on the end-effector location. For example, if the temperature of ambient is supposed to be one determined factor, the polynomial function can be rewrite as:

$$\begin{aligned}{}^{G}X_{E} &= a_{1}d_{x} + a_{2}d_{y} + a_{3}d_{z} + a_{4}{d_{x}}^{2} + a_{5}{d_{y}}^{2} + a_{6}{d_{z}}^{2} + a_{7}d_{x}d_{y} + a_{8}d_{y}d_{z} \\ &\quad + a_{9}d_{x}d_{z} + a_{10} + {}^{G}X_{B} + a_{11}T + a_{12}T^{2}\end{aligned} \tag{4.10}$$

$$\begin{aligned}{}^{G}Y_{E} &= b_{1}d_{x} + b_{2}d_{y} + b_{3}d_{z} + b_{4}{d_{x}}^{2} + b_{5}{d_{y}}^{2} + b_{6}{d_{z}}^{2} + b_{7}d_{x}d_{y} + b_{8}d_{y}d_{z} \\ &\quad + b_{9}d_{x}d_{z} + b_{10} + {}^{G}Y_{B}\end{aligned} \tag{4.11}$$

$$\begin{aligned}{}^{G}Z_{E} &= c_{1}d_{x} + c_{2}d_{y} + c_{3}d_{z} + c_{4}{d_{x}}^{2} + c_{5}{d_{y}}^{2} + c_{6}{d_{z}}^{2} + c_{7}d_{x}d_{y} + c_{8}d_{y}d_{z} \\ &\quad + c_{9}d_{x}d_{z} + c_{10} + {}^{G}Z_{B} \quad .\end{aligned} \tag{4.12}$$

On the other hand, it is possible to choose and eliminate non-effect terms by applying *Stepwise* regression which will be described in details later in Section 4.3.2.This is a significant advantage of this model. Since the increasing complexity of the model normally results in a reduction of prediction error and a rising of cost in computation. However, it does not imply that the most complex model is the best model, since the corresponding computation and implemental cost will increase as well.

4.2 Data Acquisition

As discussed in the beginning of this chapter, a number of correspondences between the joint inputs and the positions of end-effector need to be collected. Compared with the joint inputs that can be easily recorded, the positions of the end-effector are more challenging to be determined. Many efforts have been made to improve the accuracy and reliability of the end-effector position measurement. In [43], the problem was solved using a local illumination system by lighting the end-effector with an optical fiber. In a similar way, [44] employed two infrared LEDs that were mounted on the bottom of a robot to measure the end-effector positions. In the microrobotic fiber characterization platform used in this thesis, a gripper jaw acts as the end-effector for fiber manipulation, as depicted in Figure 4.1. However, it is difficult and even impossible to identify the same reference point on the jaw every time, and thus the gripper jaw position measurements are not accurate and reliable. Motivated by the vision system calibration, a chessboard pattern is attached to the gripper for indirectly measuring the end-effector position in the global frame [45]. Since the spatial relationship between the chessboard frame$\{C\}$ and the jaw frame $\{J\}$ is stationary, the kinematics from the base frame $\{B\}$ to the jaw frame $\{J\}$ can be simplified to the kinematics from the base frame $\{B\}$ to the chessboard frame$\{C\}$.

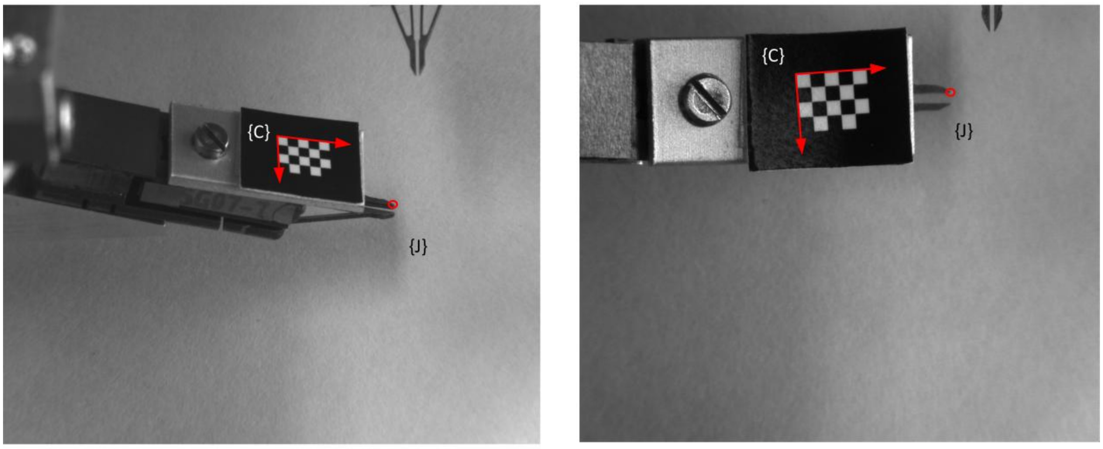


Figure 4.1 Spatial relationship between the chessboard and the gripper jaw from top and side view

This approach has the following advantages:

- The chessboard pattern includes high contrast line and corner features which are easily to be detected.
- It is straightforward and convenient to set the correspondence between the corner features of the physical chessboard and the image.
- Ready-to-use functions, being able to measure features from images with sub-pixel accuracy, are available in both *OpenCV* and *Matlab* Library.

In the beginning, only one extracted corner that represents the end-effector is used to calculate the position in the workspace. Ideally, if the movement of the chessboard keeps constant, the spatial distances between corners in two images should have the same value. However, as shown in Table 4-1, the standard deviations of the measurements are 0.207 pixels and 0.172 pixels on *x* and *y* directions, respectively. The results imply that the repeatability of the measurement needs to be improved. Thus, a virtual point $[P_v^x, P_v^y]$ that is calculated from the locations of all the chessboard inner corners are used to represent the end-effector:

$$P_v^X = \frac{\sum_{i=1}^{N} P_i^X}{N} \quad P_v^Y = \frac{\sum_{i=1}^{N} P_i^Y}{N} \; . \tag{4.13}$$

where, $[P_i^x, P_i^y]$ is the location of inner corner *i* in the 2D image frame. Since the small measurement errors cancel each other by taking the mean value of several feature points, the stand deviations are reduced to 0.087 pixels and 0.045 pixels on *x* and *y* direction, respectively.

Table 4-1 Comparison between one point method and virtual point method

Measurement Method	**Standard Deviation of Measurement**	
	X direction (Pixels)	***Y direction(Pixels)***
One Point Method	0.207	0.172
Virtual Point Method	0.087	0.045

Based on the previous knowledge, a program that can automatically record the joint inputs and the positions of the end-effector has been developed. The general procedures based on the software are as following:

Commands Generation: The trajectory consists of 100 random movements in X-Y-Z directions which are generated by the computer automatically. One important point here needs to be noticed, in order to gain a proper kinematic model, it is necessary to examine that the generated commands have as many different robot configurations as possible.

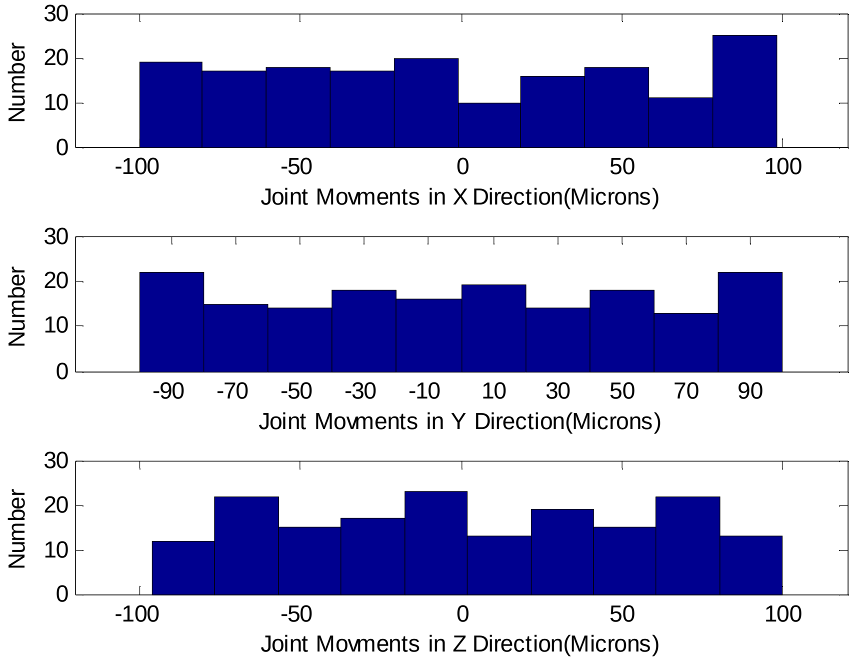


Figure 4.2 Distribution of the sampled joint movements for the calibration.

Figure 4.2 shows that the acquired commands are various enough within the cameras' measurement space for the kinematic calibration. The generated command list will be stored in a text file.

Actuator Driven and Image acquisition: The commands list in the text file is read line by line and transformed into input voltage signals for actuator through D/A converters. Since the low sampling frequency of the camera, it is impossible to measure the micromanipulator movements which has higher frequency without aliasing based on the Nyquist–Shannon sampling theorem. Hence, the micromanipulator and cameras are arranged with the following loop sequence: the micromanipulator executes the command and then waits 30 seconds for image acquisition by the cameras. The acquired images are named in either *top_#.tiff* or *side_#.tiff* to indicate the source camera and the chronological sequence.

Corner Feature Extraction and Virtual Point Calculation: The corner features of the chessboard on every image are extracted through *OpenCV* functions: *cv2.findChessboardCorners* and *cv2.cornerSubPix* which provide sub pixel accuracy. The virtual point location is then calculated based on Equation 4.13 with 12 corner feature positions and stored in MAT-files: *virtual_top.mat* and *virtual_side.mat*. As discussed in Section 3.2, the 3D location of the end-effector (virtual point) can be reconstructed based on the position information that is extracted from top and side views.

4.3 Parameter Identification

In Section 3.1 and Section 3.2, the candidate parametric models have been derived, and a number of correspondences includes the joint inputs and the positions of end-effector have been collected. The search for the best model now becomes a problem of determining the parameter vector $\boldsymbol{\lambda}$. The problem can be stated as:

Given:

1) Candidate model:
$[{}^{G}X_{E}, {}^{G}Y_{E}, {}^{G}Z_{E}]^{T} = f([{}^{G}X_{B}, {}^{G}Y_{B}, {}^{G}Z_{B}]^{T}, \boldsymbol{\Theta}, \boldsymbol{\lambda})$
2) Correspondence sets:
Actuator inputs: $\boldsymbol{\Theta}$
End-effector frame positions: $[{}^{G}X_{E}, {}^{G}Y_{E}, {}^{G}Z_{E}]^{T}$
Base frame positions: $[{}^{G}X_{E}, {}^{G}Y_{E}, {}^{G}Z_{E}]^{T}$
Compute:

Parameter Vectors: $\boldsymbol{\lambda}$

Correspondingly, this section presents two common approaches for the parameter identification in the matrix model and polynomial model, respectively. In Section 4.3.1 the result of linear lest square regression for the matrix model is introduced. Moreover, the principle and algorithm of stepwise regression for polynomial model is discussed in Section 4.3.2.

4.3.1 Linear Least Square Regression

In Section 3.1.2, the linear least square regression method is introduced to solve the unknown camera matrix **P** in the linear Equation 3.7. In a same manner, the unknown parameter matrix **R** in the matrix model (Equation 4.5) can be estimated. Since the algorithm has already been discussed in details previously, here only the estimation result of Equation 4.5 is presented. The least-squares estimate for the matrix **R** is:

$$\mathbf{R} = (\mathbf{\Theta}^{\mathbf{T}}\mathbf{\Theta})^{-1}\mathbf{\Theta}^{\mathbf{T}}({}^{\mathbf{G}}\mathbf{P}_{\mathbf{E}} - {}^{\mathbf{G}}\mathbf{P}_{\mathbf{B}})\,, \tag{4.14}$$

where, the form $(\mathbf{\Theta}^{\mathbf{T}}\mathbf{\Theta})^{-1}\mathbf{\Theta}^{\mathbf{T}}$ is the Moore–Penrose pseudo inverse of the Joint variables vector. ${}^{\mathbf{G}}\mathbf{P}_{\mathbf{E}}$ and ${}^{\mathbf{G}}\mathbf{P}_{\mathbf{B}}$ are the location vectors of the end-effector frame $\{E\}$ and the base frame$\{B\}$ in the vision frame $\{V\}$, respectively.

4.3.2 Stepwise Regression

As seen in Equation 4.6, the relationship between the actuator inputs and the end-effector positions is 2nd order nonlinear. Furthermore, since the model is not based on physical modeling, the significance of each term (actuator inputs) to the output (end-effector position) is unknown. Thus, an algorithm that is not only able to estimate the values of coefficients but also able to eliminate the least significant terms is needed.

The *stepwise* algorithm, as an automatic procedure for statistic model selection, is able to meet the aforementioned requirements [46]. Since the candidate variables will be tested one by one for the statistical significance with *F*-tests to decide variables to be included or excluded, the *stepwise* algorithm is especially suitable for a model which has a large number of potential explanatory variables. The method proceeds as follow:

1) Calculate the correlation of all the variables with output. Select the highest correlated one as the first variable to enter the test. If the variable can pass the overall *F*-test (p-values below an entrance tolerance), it will be retained.

2) Calculate the partial correlation of the rest variables. Choose the one with the highest partial correlation to enter the test. Similar with Step 1, this term will be retained if it can pass the overall *F*-test.

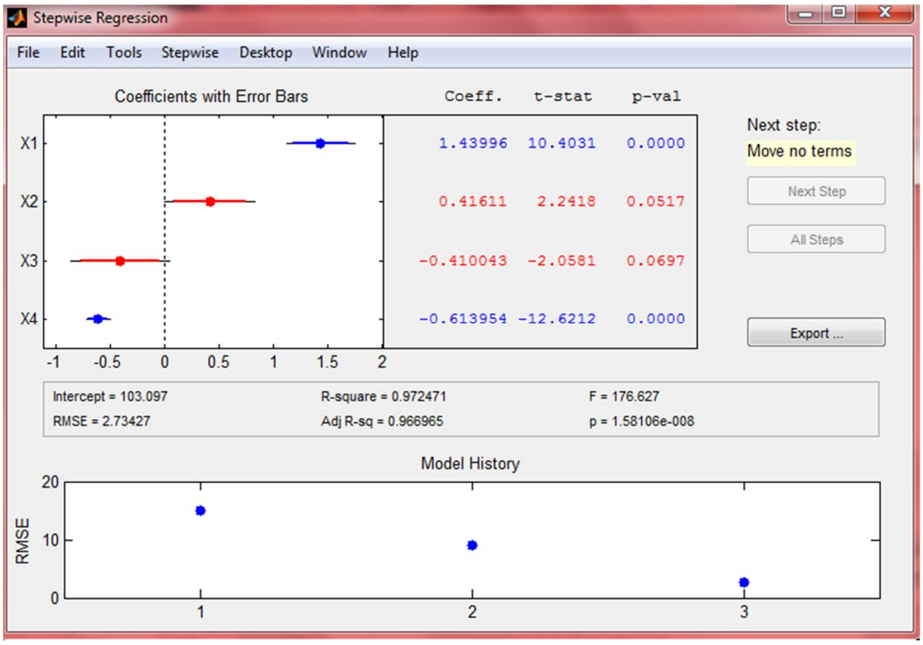
3) Repeat Step 2 until all the potential variables have been tested. All the parameters are iteratively refined based on their statistical significance.

This algorithm is provided as a ready-to-use function in Matlab, Statistic toolbox. Let us take X_E as an example to illustrate the general procedure to estimate the parameters by *stepwise* algorithm in Matlab:

1) Rearrange the vectors of end-effector positions, actuator inputs, 2nd order of actuator inputs as column vectors.

$$\begin{bmatrix} {}^{G}X_{E1} \\ \cdots \\ {}^{G}X_{Em} \end{bmatrix} = a_1\begin{bmatrix} d_{x1} \\ \cdots \\ d_{xm} \end{bmatrix} + a_2\begin{bmatrix} d_{y1} \\ \cdots \\ d_{ym} \end{bmatrix} + a_3\begin{bmatrix} d_{z1} \\ \cdots \\ d_{zm} \end{bmatrix} + a_4\begin{bmatrix} d_{x1}{}^{2} \\ \cdots \\ d_{ym}{}^{2} \end{bmatrix} + a_5\begin{bmatrix} d_{y1}{}^{2} \\ \cdots \\ d_{ym}{}^{2} \end{bmatrix} + a_6\begin{bmatrix} d_{z1}{}^{2} \\ \cdots \\ d_{zm}{}^{2} \end{bmatrix} \\ + a_7\begin{bmatrix} d_{x1}d_{y1} \\ \cdots \\ d_{xm}d_{ym} \end{bmatrix} + a_8\begin{bmatrix} d_{y1}d_{z1} \\ \cdots \\ d_{ym}d_{zm} \end{bmatrix} + a_9\begin{bmatrix} d_{x1}d_{z1} \\ \cdots \\ d_{xm}Z_{zm} \end{bmatrix} + a_{10}\begin{bmatrix} 1 \\ \cdots \\ 1 \end{bmatrix} + \begin{bmatrix} {}^{G}X_{B1} \\ \cdots \\ {}^{G}X_{Bm} \end{bmatrix}. \qquad (4.15)$$

2) Launch the interactive tool for *stepwise* regression (Figure 4.3).



```
            Coeff.     t-stat    p-val
X1         1.43996    10.4031   0.0000
X2         0.41611     2.2418   0.0517
X3       -0.410043    -2.0581   0.0697
X4       -0.613954   -12.6212   0.0000
```

Next step:
Move no terms

Intercept = 103.097 R-square = 0.972471 F = 176.627
RMSE = 2.73427 Adj R-sq = 0.966965 p = 1.58106e-008

Figure 4.3 Matlab interface for stepwise regression

3) Export the term *beta* which includes the parameter vectors of the model and term *stats* which includes the intercept.

4.4 Compensation Implementation

This section describes the implementation details of the calibrated kinematic model developed in this work. As discussed in Section 2.3.2, an open loop control strategy has been selected. Figure 2.5 depicts the architecture of the control scheme. The objective is to pre-shape the actuator inputs based on the inverse kinematic model. The problem can expressed as:

Given:

1) Forward Kinematic:

$$[{}^{G}X_{E},\ {}^{G}Y_{E},\ {}^{G}Z_{E}]^{T} = f\,([{}^{G}X_{B},\ {}^{G}Y_{B},\ {}^{G}Z_{B}]^{T},\ \boldsymbol{\Theta}\ ,\ \boldsymbol{\lambda}\ )$$

2) Identified Model Parameter: $\boldsymbol{\lambda}$

3) Desired End-effector Position: $[{}^{G}X_{E},\ {}^{G}Y_{E},\ {}^{G}Z_{E}]^{T}$

Compute:

The joint inputs: $\boldsymbol{\Theta}$

4.4.1 Matrix Compensator

In the matrix model, it is straight forward to formulate the compensator based on the inverse kinematics. Moreover, since the parameter matrix **R** is a 3×3 square matrix, the inverse kinematic matrix is computed as:

$$\begin{aligned}\mathbf{R}^{-1} &= \frac{1}{\det(\mathbf{R})}\,adj(\mathbf{R})\\ &= \frac{1}{|\mathbf{A}|}\begin{bmatrix} \begin{vmatrix} r_{22} & r_{23}\\ r_{32} & r_{33}\end{vmatrix} & \begin{vmatrix} r_{13} & r_{12}\\ r_{33} & r_{32}\end{vmatrix} & \begin{vmatrix} r_{12} & r_{13}\\ r_{22} & r_{23}\end{vmatrix}\\ \begin{vmatrix} r_{23} & r_{21}\\ r_{33} & r_{31}\end{vmatrix} & \begin{vmatrix} r_{11} & r_{13}\\ r_{31} & r_{33}\end{vmatrix} & \begin{vmatrix} r_{13} & r_{11}\\ r_{23} & r_{21}\end{vmatrix}\\ \begin{vmatrix} r_{21} & r_{22}\\ r_{31} & r_{32}\end{vmatrix} & \begin{vmatrix} r_{12} & r_{11}\\ r_{32} & r_{31}\end{vmatrix} & \begin{vmatrix} r_{11} & r_{12}\\ r_{21} & r_{22}\end{vmatrix} \end{bmatrix},\end{aligned} \tag{4.16}$$

where,$r_{ij,\ (i,j=1\ldots3)}$ are the elements in the parameter matrix **R**. Therefore, the matrix compensator based feedforward control is depicted in the Figure 4.4.

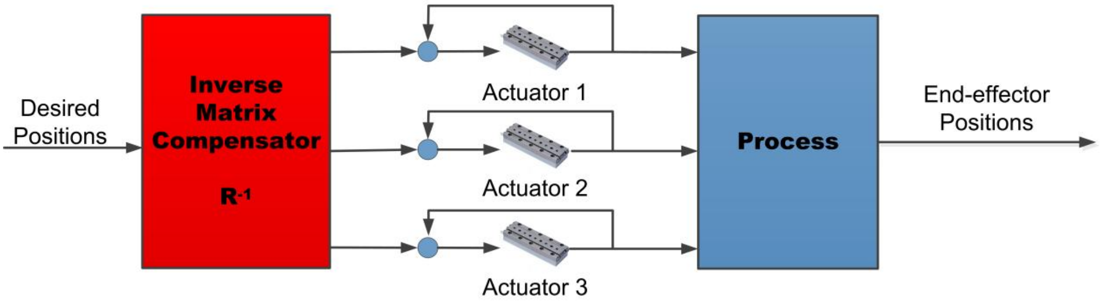


Figure 4.4 The matrix compensator based feedforward control

4.4.2 Optimal Compensator

Since the polynomial model consists of 2nd order terms, it is obvious that the inverse kinematic equations cannot be implicitly expressed as a function of the desired end-effector position $[{}^{G}X_{E},\ {}^{G}Y_{E},\ {}^{G}Z_{E}]^{T}$, as described in Section 4.1.3. A parameter optimization tool is needed to find a solution $\mathbf{\Theta}$, so that the nonlinear function:

$$\mathrm{F}=f\,([{}^{G}X_{B},\ {}^{G}Y_{B},\ {}^{G}Z_{B}]^{T},\ \mathbf{\Theta}\ ,\ \mathbf{A},\ \mathbf{B},\ \mathbf{C})-[{}^{G}X_{E},\ {}^{G}Y_{E},\ {}^{G}Z_{E}]^{T}, \tag{4.16}$$

could be converged. Since the number of unknowns in $\mathbf{\Theta}$ is equal to the number of equations, the problem is categorized as an equality constrained optimization problem.

Several well studied algorithms are suitable to handle such problems, including Gauss-Newton method, Levenberg-Marquardt method (also called trust region method), QR decomposition and Singular value decomposition (SVD). In general, all those methods first approximate the nonlinear model to be linear, and refine the parameters after several iterations. The Gauss-Newton method and Levenberg-Marquardt method belong to the same principle which starts with an initial value and then move to the certain direction within iterations to minimize the normal function. However, the difference between two methods is that the Levenberg-Marquardt algorithm is able to handle divergence when the shift vector is far from the correct direction. On the other hand, the QR decomposition and Singular value decomposition belong to the matrix computation approach that does not involve forming the normal equations.

Since ready-to-use functions are available in *Matlab, Optimization toolbox*, Gauss-Newton method and Levenberg-Marquardt method are selected in this thesis work. The procedure is illustrated as following:

1) Write an M-file object function $\mathbf{F}(\mathbf{\Theta})=0$:

$$\begin{aligned}&a_1d_x+a_2d_y+a_3d_z+a_4d_x{}^2+a_5d_y{}^2+a_6d_z{}^2+a_7d_xd_y+a_8d_yd_z\\&+a_9d_xd_z+a_{10}+{}^{G}X_B-{}^{G}X_E=0\end{aligned}\quad , \tag{4.17}$$

$$\begin{aligned}&b_1d_x+b_2d_y+b_3d_z+b_4d_x{}^2+b_5d_y{}^2+b_6d_z{}^2+b_7d_xd_y+b_8d_yd_z\\&+b_9d_xd_z+b_{10}+{}^{G}Y_B-{}^{G}Y_E=0\end{aligned}\quad , \tag{4.18}$$

$$\begin{aligned}&c_1d_x+c_2d_y+c_3d_z+c_4d_x{}^2+c_5d_y{}^2+c_6d_z{}^2+c_7d_xd_y+c_8d_yd_z\\&+c_9d_xd_z+c_{10}+{}^{G}Z_B-{}^{G}Z_E=0\end{aligned}\quad . \tag{4.19}$$

Then substitute the parameter vectors [**A**, **B**, **C**] and the end-effector positions $[X_E,\ Y_E,\ Z_E]$ with identified values and desired values respectively. Finally, store the M-file as *objfun.m*.

2) Invoke the optimization routine with first an initial guess, and then call the function *fslove*.

3) After several iterations, if the object function is converged to a solution, the solution values will be returned. Those values will be later used as the input of the actuator.

Heretofore, the optimal compensator based feedforward control is formulated as shown in Figure 4.5.

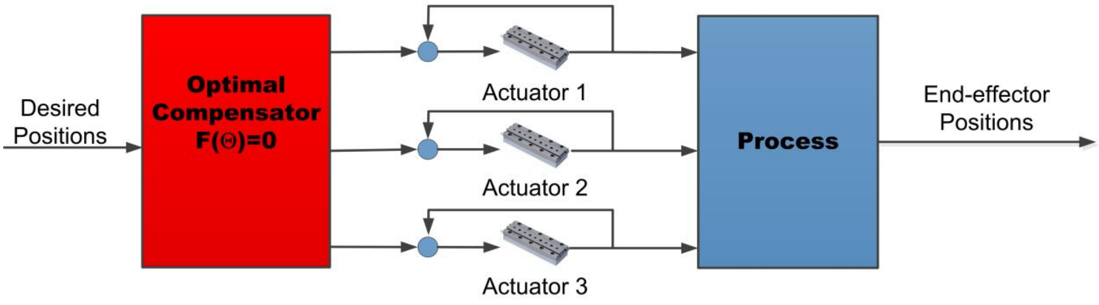


Figure 4.5 The optimal compensator based feedforward control

4.5 Summary

This chapter proposes a machine vision based generic calibration procedure for serial type micromanipulators. Firstly, two typical models: matrix model and the polynomial kinematic model are described. Also, with the aforementioned knowledge in Section 3 , the joint inputs and the corresponding positions of end-effector has been collected. The parameters of the kinematic model are then determined through linear least square regression and *stepwise* regression algorithm, respectively. Finally, the compensation schemes are then implemented to formulate matrix compensator and optimal compensator, based on the optimization of identified kinematic models. The aforedescribed four step calibration approach acts as one of the main contributions of this thesis work.

5. Case Studies and Results

In this chapter, the proposed calibration procedure is experimentally verified on the microrobotic fiber characterization platform that is introduced in Section 2.4. Section 5.1 presents the stereo-vision system calibration. Section 5.2 demonstrates the kinematic calibration results of two models for the Micromanipulator 1 in the microrobotic fiber characterization platform. Section 5.3 introduces the criteria to characterize the micromanipulator, and then presents the performance of micromanipulators before and after calibration.

5.1 Stereo-Vision System Calibration

During the stereo-vision calibration, a 25×25mm dot grid from Edmund Optics (Figure 5.1) is used as a microscale calibration target and the specifications are listed in Table 5-1.

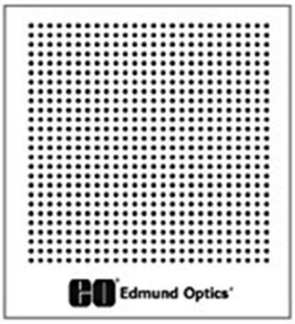


Figure 5.1 Dot grid for vision system calibration

The dot grid is placed at non-coplanar poses and under the field of view (FOV) for both side and top cameras. Then, a group of images is taken for at least six poses, and a python script is written for the automatic dot location measurements. Consequently, the points x and x' in Equation 3.17 are obtained, and used to recover the intrinsic and extrinsic coefficients in the parametric model.

Table 5-1 Specifications of dot grid target(Edmund Optics)

Type	Chrome on Glass
Dimensions (inches)	2 x 2
Pattern Size (mm)	25 x 25
Thickness (mm)	1.5

Dot Diameter (mm)	0.0625
Dot Diameter Tolerance (mm)	±0.002
Dot Spacing (mm)	0.125
Dot Spacing Tolerance (mm)	± 0.001 Center to Center, ± 0.004 Grid Corner to Corner
Overall Accuracy (mm)	±0.001
Surface Accuracy (λ)	4 - 6 per 25.4 Area
Surface Quality	40-20
Substrate	Soda Lime Float Glass
Optical Density OD	>3.0
Coating	Reflective First Surface Chromium Rabs = 50% ± 5% @ 550 nm

Following the vision system calibration procedure in Chapter 3, the camera matrixes **P** and $\mathbf{P}'$ for the top and side view cameras are computed as:

P = [1.49e+04 -5.47e+02 7.09e+02 1.61e+05

4.30e+01 1.17e+04 8.83e+03 7.93e+04

-6.28e-03 -5.67e-01 8.23e-01 3.24e+02];

$\mathbf{P}'$ = [1.28e+04 1.45e+02 -4.54e+02 8.41e+04

-9.99e+011.28e+04 5.45e+02 4.13e+04

8.51e-02 -1.52e-04 9.96e-01 2.84e+02].

According to Equation 3.19, the reprojection errors of top and side camera are 0.090917014 pixels and 0.054174204 pixels, respectively.

5.2 Micromanipulator Calibration

As described in Section 4.2 , a designed trajectory command including 170 random movements was used as an input to Micromanipulator 1. Meanwhile, the images including a chessboard pattern from both side and top views were grabbed. Based on the calibrated vision system, the corresponding locations of the end-effectors were then collected. Since the parameter estimation algorithm in used is based on the prediction error minimization of training sets, it is prone to the risk of over fitting. In order to avoid the over fitting, an independent sample of data from the same population as the training data is normally selected as validation data. In this test, the first 110 samples among the 170 collected samples are used for calibration, and the rest 60 samples are used for model validation.

5.2.1 Results of Kinematic Calibration

Following the matrix model based calibration procedure, as described in Chapter 4 , the coefficients of the matrix model are estimated as:

$$\mathbf{R} = \begin{bmatrix} 1.000885 & 0.000225 & 0.002571 \\ -0.000409 & 0.999812 & 0.003136 \\ -0.000005 & -0.001111 & 1.002641 \end{bmatrix}.$$

On the other hand, following the polynomial model based calibration procedure, as described in Chapter 4 , the coefficients of the polynomial model are estimated as:

$$\mathbf{A} = [1.000893 \quad 0 \quad 0.002475 \quad 0 \quad -4.74e-08 \quad 0 \quad 0 \quad 0 \quad 0],$$

$$\mathbf{B} = [0 \quad 0.999927 \quad 0 \quad 0 \quad 0 \quad 0 \quad 7.11e-07 \quad 0 \quad 0],$$

$$\mathbf{C} = [0.000874 \quad 0 \quad 1.000775 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0].$$

It should be noted that some coefficients show a zero value. Such terms are eliminated in the polynomial kinematic model, since their static significances are low.

5.2.2 Prediction Error of Calibrated Model

So far, the parametric model to represent the kinematic of micro *X*-*Y* table is derived. The next question is whether the real world process is properly and precisely described by the mathematic model. As illustrated in Figure 5.2 , the discrepancy between the predicted location and the measured location with the same input signal is termed as prediction errors. The prediction error represents the degree of how fit between the model’s simulated and measured output [47]. Ideally, the best model has a zero prediction error which is impossible in practice. On the other hand, the micromanipulation tasks typically allow a tolerance margin. Hence, the model is considered good enough if the prediction errors are all inside the tolerance band.

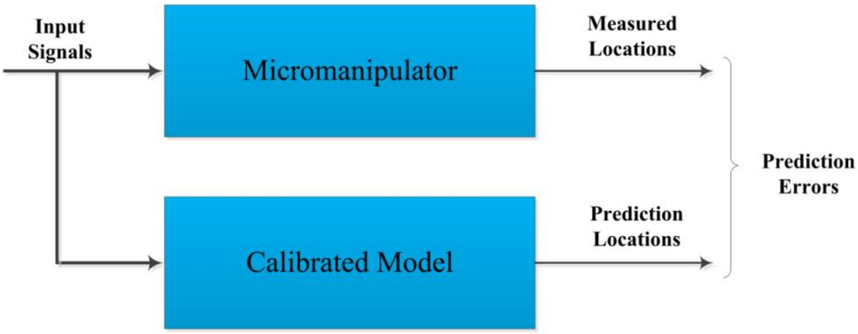


Figure 5.2. A block diagram to illustrate prediction errors

The prediction error is expressed as the spatial distance between the predicted poses and the desired pose, and it can be computed as follows:

$$PE = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2}\ , \tag{5.1}$$

where, (x_i, y_i, z_i) is the actual positions measured by the vision system, and ($\hat{x}_i, \hat{y}_i,$ $\hat{z}_i$) are the corresponding positions predicted with the calibrated kinematic model.

As mentioned above, in the first test, the trajectory command including 170 random movements with different configurations was used as an input to the micromanipulator, and the corresponding locations of the end-effector in the vision frame were collected. In the measurement set, 110 points are used for the calibration and 60 points for the model validation. The prediction errors of two calibrated model for 60 validation points are computed and illustrated in

Figure **5.3** and Figure 5.4. As shown in

Figure **5.3**, the experimental results demonstrate the maximum position error of the matrix model is smaller than 8.35 micrometers, the mean value and standard deviation of the position error are approximately 4.58 micrometers and 2.50 micrometers, respectively.

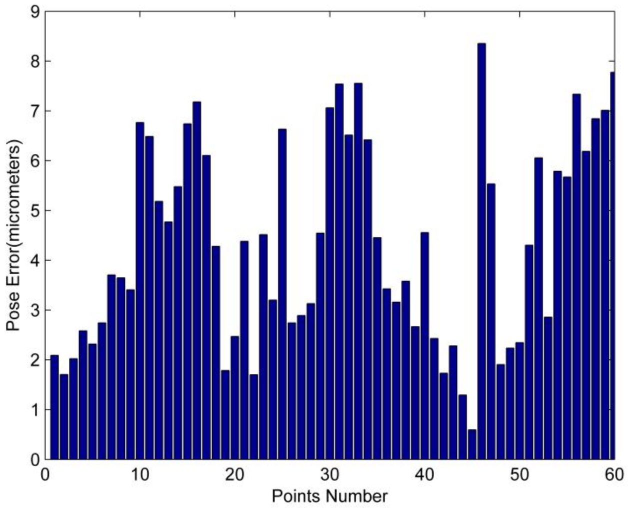


Figure 5.3 The prediction error of matrix model for 60 validation points

Figure 5.4 shows that the maximum position error of the polynomial model is smaller than 5.28 micrometers, the mean value and standard deviation of the position error are approximately 2.59 micrometers and 1.32 micrometers, respectively.

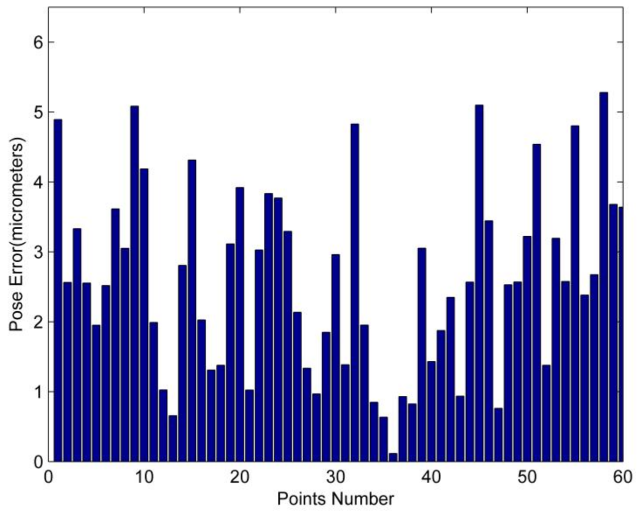


Figure 5.4. The prediction error of polynomial model for 60 validation points

Since the collected data contain measurement noise which has a same order of magnitude with pose error, it is understandable that the prediction error can only be reduced to 2.59 micrometers.

5.3 System Characterization and Experiment Verification

The aims of the system characterization experiments are to verify the validity of the proposed calibration approach, compare the system performance with and without calibration. Micromanipulator 1 in the microrobotic fiber characterization platform is selected as an example for studies in this section. The micromanipulator is examined with two tests: the pose accuracy test and the decoupling test. In both tests, errors exist when the measured displacements are compared with desired displacement. It should be noted that those errors are not only from the modeling imperfectness but also from the inaccuracy of the measurement system and the limited resolution of the actuator input.

Heretofore, two widely adopted standards exist- ISO 9283 and ANSI/RIA R15.05-1- to evaluate the performance of an industrial robot. The international standard ISO 9283 was published by International Organization for Standardization (ISO) in 1998 and includes the performance criteria and related test methods for an industrial manipulator. The ANSI/RIA R15.05-1-1990 is an American national standard that defines methods for the static performance evaluation of industrial robots. Moreover, those two standards are almost identical in content, whereas there are slight differences in the terminology and performance evaluations. Since no similar standard exists for micromanipulator performance evaluations, even though above standards are both for industrial macro-robots, the performance criteria used in the thesis are based on the ISO 9283 standards.

5.3.1 Pose Accuracy Test

The ability of the calibrated micromanipulator to accurately arrive the desired poses is studied by driving the micromanipulator to different locations inside the workspace. In the pose accuracy experiments, the micromanipulator is commanded to move to 15 random locations, and the corresponding displacements are measured by the vision system described in Section 3.

According to the ISO 9283 standard, the pose accuracy is expressed as the standard deviation of the differences between the desired pose and the average of arrival poses in all the directions, and it can be computed as follows:

$$AP_p = \sqrt{(\bar{x} - x_d)^2 + (\bar{y} - y_d)^2 + (\bar{z} - z_d)^2} \quad , \tag{5.2}$$

where, $\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$, $\bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$, and $\bar{z} = \frac{1}{n}\sum_{j=1}^{n} z_j$ are the average of arrival pose along x, y and z axis,[$x_{d,}$ $y_{d,}$ z_d] is the desired pose of the end-effector.

Table 5-2 compares the variances between the average of arrival poses measured by the vision system and the desired poses of the micromanipulator without calibration. The pose accuracy is computed according to Equation 5.2.

Table 5-2 Pose error of the micromanipulator (without calibration)

No	*Desired Pose(µm)*	*Average Arrival Pose(µm)*	*Variance(µm)*	*Pose Accuracy(µm)*
1	[2000,0,0]	[1999.1,2.4,0.8]	[0.9,2.4,0.8]	2.69
2	[8000,0,0]	[8001.4,-3.2,0.1]	[1.4,-3.2,0.1]	3.49
3	[0,2000,0]	[0.6,2002.4,-1.9]	[0.6,2.4,-1.9]	3.11
4	[0,8000,0]	[1.7,8004.5,10.5]	[1.7,4.5,10.5]	11.55
5	[0,0,1500]	[0.1,9.3,1496.4]	[0.1,9.3,3.6]	9.97
6	[0,0,3000]	[2.8,11.2,2979.5]	[2.8,11.2,- 20.5]	23.52
7	[1000,1000,600]	[1003.9,1000.1,596.7]	[3.9,0.1,3.3]	5.11
8	[2000,2000,300]	[2000.2,2003.1,299.2]	[0.2,3.1,0.9]	3.23
Average accuracy(µm)				7.83

Table 5-3 and Table 5-4 list the pose error of the micromanipulator with the matrix compensator and the optimal compensator, respectively. The maximum pose error is reduced from 23.52 µm to 9.14 µm. However, there is no significant difference between the matrix compensator and the optimal compensator. It is important to note that since the resolution of the measurement system is about ±3µm, the measurement noise is introduced in the measurement of pose under ±3 µm.

Table 5-3 Pose error of the micromanipulator (with the matrix compensator)

No	*Desired Pose(µm)*	*Average Arrival Pose(µm)*	*Variance(µm)*	*Pose Accuracy(µm)*
1	[2000,0,0]	[2002.5,-0.8,0.02]	[2.5,-0.8,0.02]	2.62
2	[8000,0,0]	[7999.3,-2.5,4.2]	[0.7,-2.5,4.2]	4.93
3	[0,2000,0]	[1.3,2000.3,2.0]	[1.3,0.5,2.0]	2.43
4	[0,8000,0]	[5.3,7999.3,5.0]	[5.3,0.7,5.0]	7.31
5	[0,0,1500]	[1.0,-1.5,1503.7]	[1.0,-1.5,3.7]	4.11
6	[0,0,3000]	[2.9,-2.7,3009.2]	[2.9,-2.7,9.2]	10.0
7	[1000,1000,600]	[999.2,995.5,600.3]	[0.8,4.5,0.3]	4.58
8	[2000,2000,300]	[2000,1994.9,300.4]	[0,5.1,0.4]	1.18
Average accuracy(µm)				4.65

Table 5-4 Pose error of the micromanipulator (with the optimal compensator)

No	*Desired Pose(µm)*	*Average Arrival Pose(µm)*	*Variance(µm)*	*Pose Accuracy(µm)*
1	[2000,0,0]	[2001.3,0.3,0.2]	[1.3,0.3,0.2]	1.35
2	[8000,0,0]	[8000.3,2.3,3.7]	[0.3,2.3,3.7]	4.37
3	[0,2000,0]	[1.3,1999.9,2.1]	[1.3,0.1,2.1]	2.47
4	[0,8000,0]	[4.1, 8001.2,3.9]	[4.1,1.2,3.9]	5.78
5	[0,0,1500]	[1.4,2.1,1500.9]	[1.4,2.1,0.9]	2.68
6	[0,0,3000]	[1.5,3.3,3008.4]	[1.5,3.3,8.4]	9.14
7	[1000,1000,600]	[999.5,993.2,601.2]	[0.5,6.8,1.2]	6.92
8	[2000,2000,300]	[1997.5,2000.1,299.5]	[2.5,0.1,0.5]	2.55
Average accuracy(µm)				4.41

5.3.2 Decoupling Test

In Section 5.3.1, the ability of the micromanipulator to move to desired locations was studied. In order to further examine the ability of the micromanipulator to produce decoupled motions, decoupling tests are carried out with and without the calibration. In the decoupling test, the micromanipulator is commanded to move first 8000 micrometers along the *x* axis, then 8000 micrometers along the *y* axis and finally 4800 micrometers along the *z* axis, as shown in Figure 5.5.

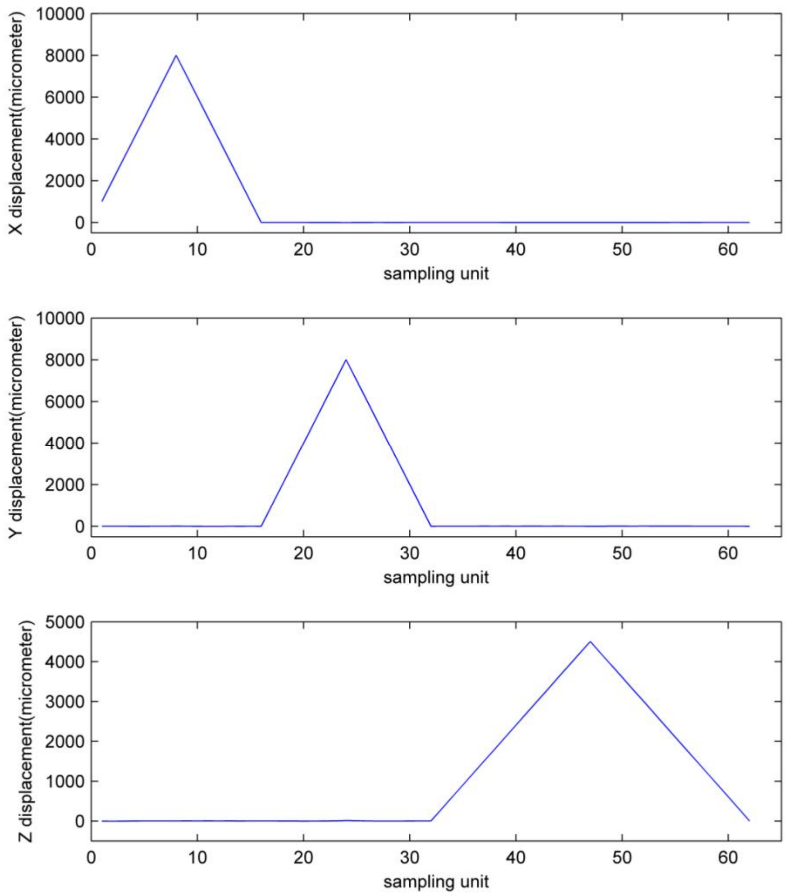


Figure 5.5 Reference signal for the micromanipulator

For the ideal case, the motions only in a single direction should not affect each other. In other words, the displacements of the micromanipulator in *y* and *z* directions should be zero when it moves along the *x* direction, the displacements of the micromanipulator in *x* and *z* directions should be zero when it moves along the *y* direction, and the displacements of the micromanipulator in *x* and *y* directions are zero when it moves along the *z* direction. However, the test results, in the real case, demonstrate that interactions exist among the motions of three axes (depicted in Figure 5.6). As shown in Figure 5.7-(a), the unwanted displacements in *x* direction, caused by motions in *y* and *z* directions, are in the range [-6.144µm,-0.104µm]. Figure 5.7-(b) implies the unwanted displacements in *y* direction, caused by motion in *x* and *z* directions, are in the range [-25.13µm,-3.21µm]. Figure 5.5-(c) depicts the unwanted displacements in *z* direction, caused by motion in *x* and *y* directions are in the range [-4.68µm, 10.55µm]. One relationship between the desired motion in single axis and the unwanted motions in other two axes can be observed in all three results: the displacements of unwanted motions are proportional to the displacement of the desired motion. The maximum unwanted displacement, about -25.13 µm, is found in *y* direction when the micromanipulator moves along the *z* direction for 4800µm.

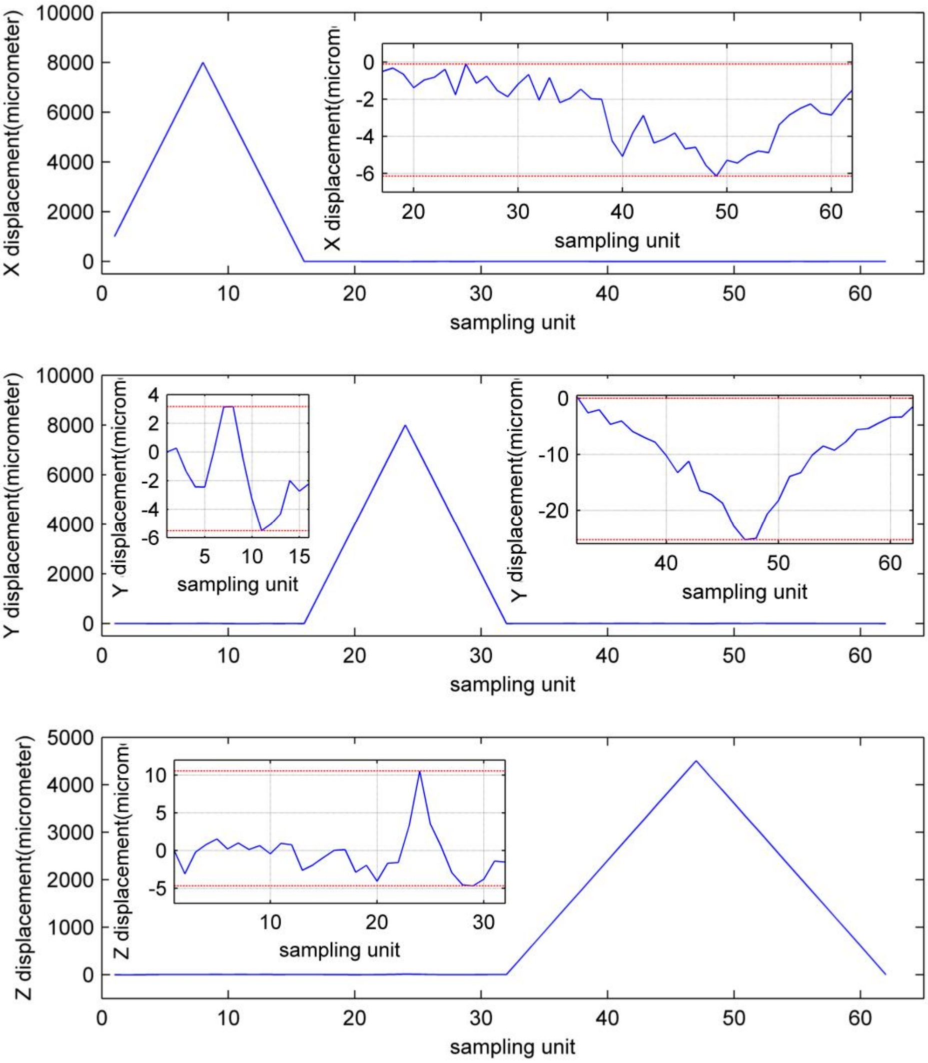


Figure 5.6 Displacements of micromanipulator for decoupled motions (without calibration)

In order to reduce the interaction among the motions in three axes, the input signals to the actuators are pre-shaped with the matrix compensator and the optimal compensator, respectively. After the compensation with the matrix compensator, the measured displacements are shown in Figure 5.7. When the micromanipulator moves along the *y* and *z* directions, the unwanted displacements in *x* direction are reduced to the range [-5.29µm, 0.042µm]. The unwanted displacement in *y* direction during the movement in *x* and *z* directions are [-5.53µm, 5.63µm], approximately 50% smaller than uncalibrated case. When the micromanipulator is driving along the *x* and *y* direction, the unwanted displacement is reduced to the range [-0.88µm, 6.069µm].

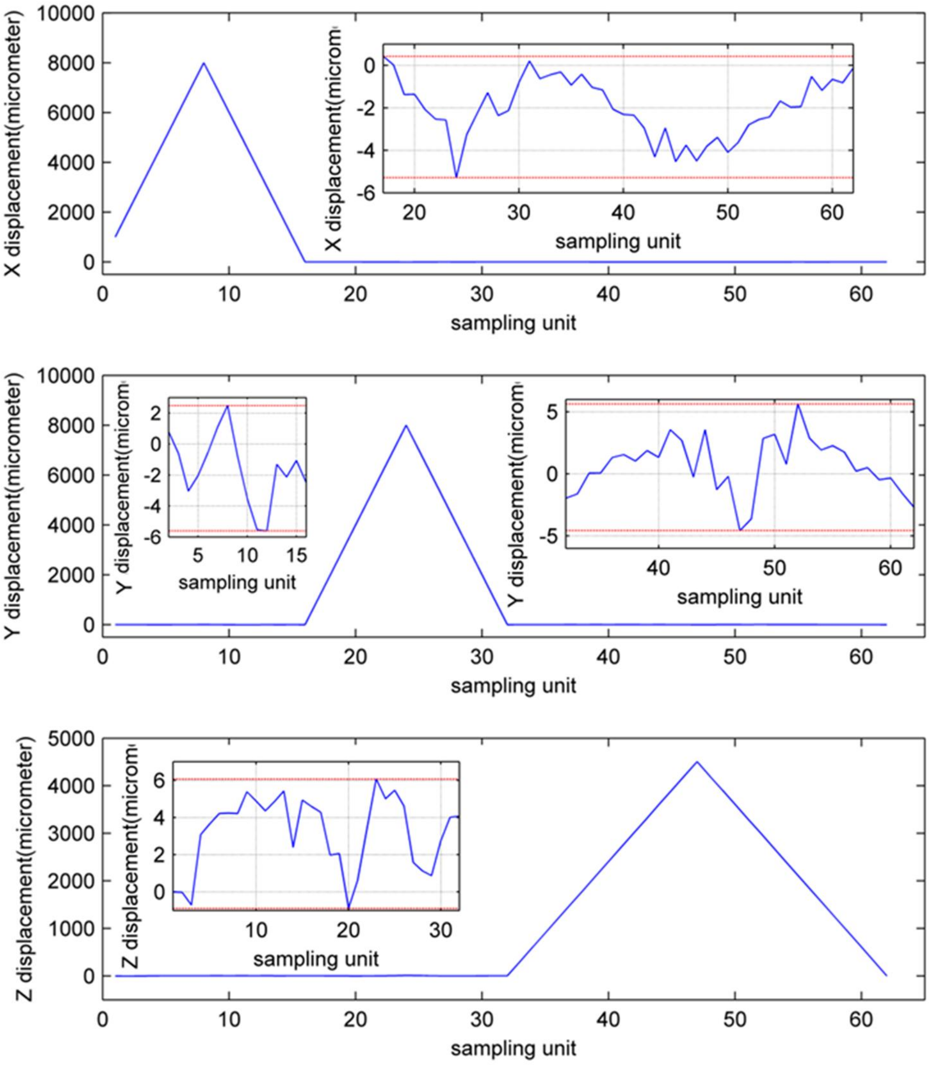


Figure 5.7 Displacements of micromanipulator for decoupled motions (with matrix compensator)

Figure 5.8 shows the measured displacements of the micromanipulator for decoupled motions, after compensated by the optimal compensator. When the micromanipulator moves along the *y* and *z* directions, the unwanted displacements in the *x* direction are reduced to the range [-4.33µm, 0.21µm]. The unwanted displacement in the *y* direction during the movement in the *x* and *z* directions are [-5.11µm, 3.64µm], approximately 60% smaller than uncalibrated case. When the micromanipulator is driving along the *x* and *y* direction, the unwanted displacement is reduced to the range [-0.71µm, 5.986µm].

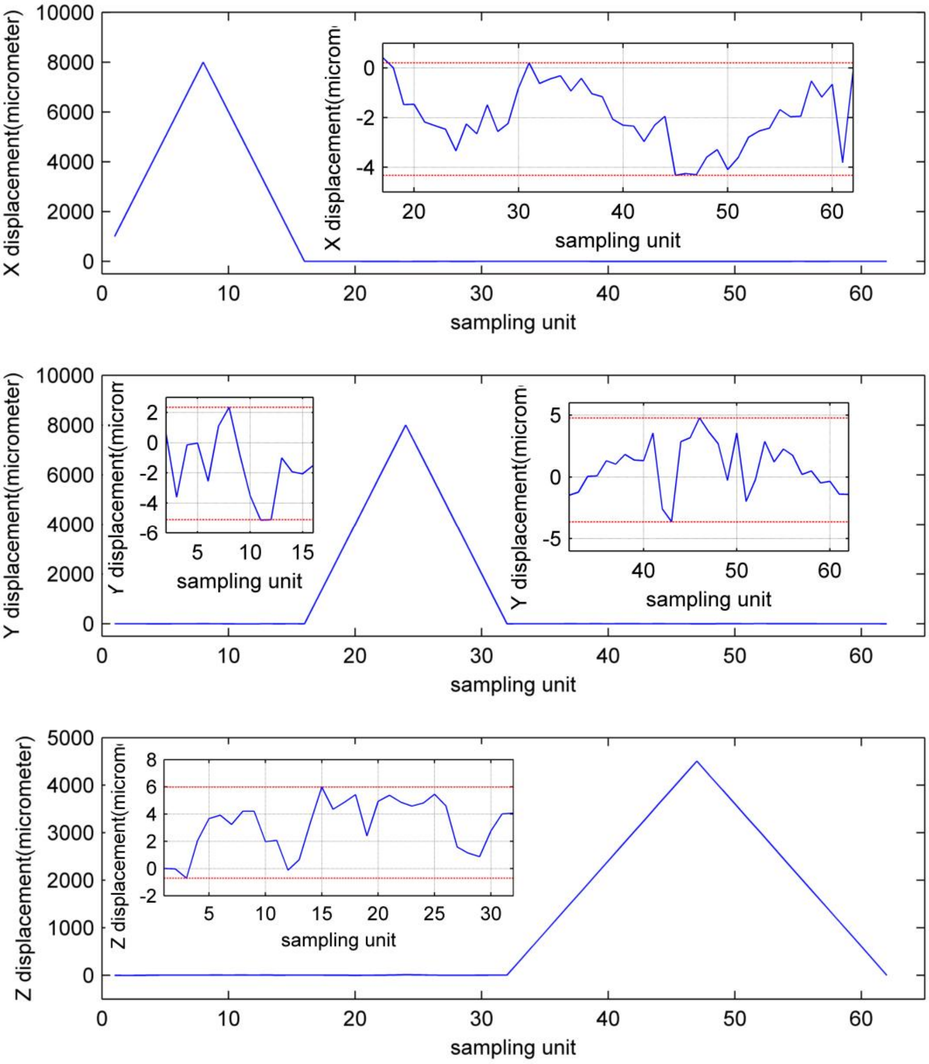


Figure 5.8 Displacements of micromanipulator for decoupled motions (with optimal compensator)

To summarize, the ability of the micromanipulator to produce decoupled motions is improved by the calibration of the micromanipulator. Table 5-5 compares the maximum undesired displacements of three axes with and without calibration. The result shows that the undesired displacements for decoupled motions are all reduced. However, the percentages of improvement are various in three axes, where the maximum is about 60% in the *y* direction and the minimum is about 13% in the *z* direction. One reason to explain the difference is that the limited resolution of measurement system, which is about ± 3μm inaccuracy. On the other hand, the undesired displacement is relative small, less than 10 μm for 8000 μm movement, except the one in the *y* direction which is -25.13 μm for 4800 μm movement. Therefore, the function of calibration is limited if the kinematic error is smaller than the measurement error. However, the result in the *y* direction demonstrates that the error larger than 10 µm can be dramatically reduced through the calibration approach.

Table 5-5Comparisons of undesired displacements with and without compensations

Direction	***Undesired Displacements***		
	Uncalibrated (µm)	***With the Matrix Compensator(µm)***	***With the Optimal Compensator(µm)***
X	[-6.14, -0.104]	[-5.29, 0.04]	[-4.33, 0.21]
Y	[-25.13, 3.21]	[-5.53,5.63]	[-5.11, 3.64]
Z	[-4.68, 10.55]	[-0.88 6.07]	[-0.71, 5.99]

On the other hand, the undesired displacement with the matrix compensator and the optimal compensator are compared. The result shows that in most directions the performance of the optimal compensator is better than the matrix compensator whereas the improvement is not obvious.

In order to further compare the performance between two compensators, an integration index Ω is introduced. The integration index is defined as the integration of undesired displacements against the sampling time:

$$\Omega = \int_{t_1}^{t_2} e dt \tag{5.3}$$

where, [t_1, t_2] is the time domain of integration. In Table 5-6, the integration index of both matrix compensator and optimal compensator are compared. It is logical to interpret the integration index Ω as energy of errors which includes both static error and dynamic error. As discussed in Chapter 2, the calibration approach can only reduce the static error. Therefore, after eliminating the static error, integration indexes including only dynamic error should be in the same order for either matrix compensator or optimal compensator which has been verified in Table 5-6. Additionally, the result does not show obviously improvement comparing the matrix compensator and the optimal compensator.

Table 5-6 Comparisons of integration index Ω

Direction	***Integration index***		
	Uncalibrated (µm × sampling unit)	***With the Matrix Compensator (µm × sampling unit)***	***With the Optimal Compensator (µm × sampling unit)***
X	120.83	92.78	90.54
Y	342.61	45.75	48.92
Z	97.84	100.54	98.04

Moreover, as listed in Table 5-7, the cost of better performance for polynomial model is the long computation time and complex model structure. Therefore, if the requirement of positioning accuracy is not rigorous, it is recommended to use the matrix model.

Table 5-7 Comparisons between matrix model and polynomial model

	Matrix model	***Polynomial model***
Model Complexity	Simple	Complex
Difficulty of Implementation	Easy(Ready to use function in several libraries)	Difficult(Ready to use function only in Matlab)
Computation Time Cost	Short	Long
Performance for Small Error (With limited measurement resolution)	Fair	Fair
Performance for Large Error (With limited measurement resolution)	Good	Very Good

5.4 Summary

In this chapter, the proposed vision-based calibration procedure is verified on the microrobotic fiber characterization platform and the performance is evaluated. After calibration, the projection errors are computed as 0.1pixel for top view and 0.06pixel for side view, and the vision measurement system can provides a measurement accuracy of ± 3µm. By utilizing such a vision measurement system, the micromanipulator is calibrated following the proposed calibration procedure, and the results demonstrate that the prediction error of the calibrated matrix model and the polynomial model are below 4.58µm and 2.59µm, respectively. Those kinematic models are then employed as the basis of feed-forward compensators for the open-loop control of micromanipulator. According to the ISO 9283 standard, decoupling and pose accuracy tests are carried out with and without the calibration. Compared with the performances of the micromanipulator before calibration, the ability of micromanipulator to produce decoupled motions is improved. In addition, the calibrated model is able to reduce the maximum pose error is reduced from 23.52 µm to 9.14 µm.

6. Conclusion and Future Work

This section includes the conclusions of this thesis and a proposal for the future work of microrobotic system calibration.

6.1 Conclusions

In this thesis, a generic vision based calibration procedure for a microrobotic system has been proposed and experimentally verified. The proposed calibration procedure includes two major issues: stereo-vision system calibration and micromanipulator kinematic calibration. For the stereo-vision system calibration, the parametric camera model, the principle of stereo-vision system, and the algorithms for 3D reconstruction have been presented in Section 3. Then, the micromanipulator kinematic calibration is summarized as a four steps procedure:

- Kinematic Modeling;
- Data Acquisition;
- Parameter Estimation;
- and Compensation Implementation.

After a brief literature review, the matrix model and the polynomial model are selected to represent the kinematics of micromanipulator. The matrix model has the advantages of simple structure and easy to implement. By considering the 2nd order terms, the matrix model evolves to the polynomial model which includes both geometric and non-geometric terms. Moreover, one important advantage of this model is the extendibility that just by adding extra terms if additional factors show significant impacts on the end-effector location. The aforementioned stereo-vision system is employed in the step of data acquisition. 3D pose information of the micromanipulator end-effector are collected. This data and corresponding actuator displacement information are then divided into two sets: estimation set and cross validation set. During the parameter estimation, the estimation set data is utilized with the parameter estimation algorithms. Evaluated by the cross validation set data, the prediction accuracy of 4.58µm and 2.59µm has been achieved by the matrix model and the polynomial model, respectively. Finally, the calibrated kinematic model is implemented in an open-loop feedforward control scheme. The result demonstrates that the pose error of micromanipulator can be reduced to below 5 µm micrometers which fulfill the requirement for the fiber manipulation. In the decoupling test, the maximum undesired displacement is observed in y direction when the micromanipulator moves along the z direction .The undesired displacement is declined from [-25.13 µm, 3.21 µm] to [-5.53 µm, 5.63 µm] with the matrix compensator. On the other hand, the optimal compensator could also reduce the undesired displacement from [-25.13 µm, 3.21 µm] to [-5.11 µm, 3.64 µm].

Compared with other documented microrobot calibration methods, the proposed procedure has following novelties:

1. Even though several machine vision based calibration have been demonstrated for 1 - 2.5D applications. It is the first time to introduce a stereo vision measurement system for 3D microrobot calibration. Moreover, the feasibility has been experimentally verified.

2. The proposed procedure is easy to use and fast to implement. Without extra instruments and operations, the undesired displacement could be reduced to an accepted level. With the developments of image acquisition instruments and image processing techniques, a better calibration result could be expected.

3. In this approach, a chessboard pattern is not only used as a calibration board for camera calibration but also used as a target object for microrobot calibration. More detailed discussions of the advantages could be found in Section 4.2.

6.2 Future Work

In the future, the continuation of thesis can be carried out from the following five aspects:

1. Due to the lack of precision dimensional metrology instrument in the sub micrometer-scale (≤0.1µm), the accuracy of the stereo-vision system is not assessed. However, this measurement is of great importance to derive a more accurate kinematic model. Therefore, the accuracy of the stereo-vision system in all the three directions should be examined.

2. The influences of the ambient variables, such as temperature, humidity and vibration is not measured and investigated. It can be studied by configuring corresponding measurement instrument in the microrobotic cell. Additionally, with the benefits of extendable, those impact factors, if necessary, can be easily added to the model.

3. The kinematic model proposed in this work is based on the assumptions that the process is time invariant. The future work can extend this static model to a dynamic model, so that it is applicable for the dynamic calibration of microrobotic system.

4. In this thesis work, the vision system is stationary, and thus the field of view is limited. However, for the large scale robotic system, the sensor fusion of multi-views should be further studied.

5. For industrial applications, the robots are usually required to be able quickly plug-in and plug out for the process chain. Hence, an agile and flexible microrobotic system calibration approach needs to be investigated and proposed.

Since the field of microrobotic is undergoing a dramatic revolution nowadays, it can be foreseen that further studies in this topic will become more and more mature. This thesis, hopefully, could serve as a building brick to pave the way for the future studies.

References

[1] P. Kallio and J. Kuncova-Kallio, "Capillary pressure microinjection of living adherent cells: challenges in automation," *Journal of Micromechatronics,* vol. 3, pp. 189-220, 2006.

[2] W.Wang, X.Liu, B. D.Gelinas and Y.Sun, "A Fully Automated Robotic System for Microinjection of Zebrafish Embryos," in *PLoS ONE,* Sep. 2007, pp. Vol.2,No.9.

[3] D. Popa, R. Murthy, M. Mitta, J. Sin and H. Stephanou, "M3-Modular Multi-Scale Assembly System for MEMS Packaging," in *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2006.

[4] K. R. Sylwester Bargiel, C. Clevy, C. Gorecki and P. Lutz, "Towards micro-assembly of hybrid MOEMS components on a reconfigurable silicon free-space micro-optical bench," *Journal of Micromechanics and Microengineering,* vol. 20, 2010.

[5] P. Saketi, A. Treimanis, P. Fardim, P. Ronkanen and P. Kallio, "Microrobotic platform for manipulation and flexibility measurement of individual paper fibers," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference,* Taipei, Oct. 2010.

[6] P. Saketi and P. Kallio, "Microrobotic Platform for Making, Manipulating and Breaking Individual Paper Fiber Bonds," in *International Symposium on Assembly and Manufacturing,* Tampere, 2011.

[7] T. Fukuda and F. Arai, "Micromanipulation and Robotic Technology," in *Technical Proceedings of the 1998 International Conference on Modeling and Simulation of Microsystems,* 1998.

[8] G. Li and N. Xi, "Calibration of a Micromanipulation System," in *Proceedings of the 2002 IEEE/RSJ intl.Conference on intelligent Robots and Systems,* EPFL,Lausane,Switzerland, October,2002.

[9] A. Kawaji, F. Arai and T. Fukuda, "Calibration for Contact Type of Micromanipulator," in *Proceeding of the 1999IEE/RSJ international Conference on Intelligent Robots and System*, 1999.

[10] Y.L.Zhang, M.L.Han, C.Y.Shee, T.F.Chia and W.T.Ang, "Self-Calibration method for vision-guided cell micromanipulation system," *Journal of Microscopy*, vol. 233, pp. 340-345, Pt 2 2009,.

[11] L. S.Mattos and D. G. Caldwell, "A Fast and Precise Micropipette Positioning System based on Continous Camera-Robot Recalibration and Visual Servoing," in *5th Annual IEEE Conference on Automation Science and Engineering*, Bangalore,India, 2009.

[12] Q. Zhou, P. Korhonen, L. Jukka and S. Sjoval, "Automatic dextrous microhandling based on a 6 DOF microgripper," *Journal of Micromechatronics,* vol. 3, pp. 359-387, 2006.

[13] H. Stone, Kinematic modeling, identification, and control of robotic manipulators, Kluwer Academic Publishers, 1987.

[14] A. Hofmann, G. Bretthauer, N. Siltala and R. Tuokko, "Evolvable Micro Production Systems:Specific Needs and Differences to Macro," in *IEEE International Symposium on Assembly and Manufacturing*, Tampere, 2011.

[15] B. J. Nelson, L. X. Dong and F. Arai, "Micro/Nanorobotics," in *Springer Handbook of Robotics*, 1st ed., Heidelberg, Springer-Verlag GmbH, 2008.

[16] M. Gauthier and S. and Régnier, Robotic micro-assembly, IEEE press, 2010.

[17] N. Chaillet and S. Régnier, Microrobotics for Micromanipulation, Wiley-ISTE, 2010.

[18] J. Nielsen, T. Tanikawa and T. Arai, "Design and analysis of a 3-DOF micromanipulator," in *IEEE International Conference on Robotics and Automation*, 1999.

[19] E. M. S. D. a. N. L. F.-P. B. Tamadazte, "CAD Model-based Tracking and 3D Visual-based Control for MEMS Microassembly," *The International Journal of Robotics Research* , vol. 29, pp. 1416-1434, September 2010.

[20] Y. Yun and Y. Li, "Optimal design of a 3-PUPU parallel robot with compliant hinges for micromanipulation in a cubic workspace," *Robotics and Computer-Integrated Manufacturing*, pp. 977-985, 2011.

[21] SmarAct GmbH, 21 06 2012. [Online]. Available: http://www.smaract.de/.

[22] S. Fatikow, A. Buerkle and J. Seyfried, "Automatic control system of a microrobot-based microassembly station using computer vision," in *Conference on Microrobotics and Microassembly,* Boston, MA, USA, 1999.

[23] B. Borovic, A. Liu, D. Popa, H. Cai and F. Lewis, "Open-loop versus closed-loop control of MEMS devices: choices and issues," *J. Micromech. Microeng,* vol. 15, pp. 1917-1924, 2005.

[24] V. Ganesan, K. Wiedmann, G. Janssen, B. Ponick and A. Mertens, "Development of Drive Systems Based on Electromagnetic Microactuators," *Industrial Electronics, IEEE Transactions on,* vol. 59, pp. 1409-1417, 2012 March.

[25] Y. Sun, D.Piyabongkam, A.Sezen and R. B.J.Nelso, "A high-aspect-ratio two-axis electrostatic microactuator with extended travel range," *Sensors and Actuators A: Physical,* vol. 102, no. 1-2, pp. 49-60, 2002.

[26] M. Rakotondrabe and P. Lutz, "Main aspects of the control issues in the Micro/Nano-World," in *ICRA,* Shanghai, 2011.

[27] Q. Zhou, "Impact of Ambient Environment on Micro Interactions," in *IEEE International Conference on Robotics and Automation*, ShangHai, 2011.

[28] A. Requicha, S. Meltzer, F. Arce, J. Makaliwe, H. Siken, S. Hsieh, D. Lewis, B. Koel and M. Thompson, "Manipulation of nanoscale components with the AFM:principle and applications," in *Proceedings of the 2001 1st IEEE Conference on Nanotechnology*, Los Angeles, 2001.

[29] S.Fatikow and U. Remobold, Microsystem technology and microrobotics, New York: Springer-Verlag Berlin Heidelberg, 1997.

[30] Y. Haddab, Y. Le Gorrec and P. Lutz, "Effects of enviormental noise on the accuracy of millimeter sized grippers in cantilever configuration and active satbilisation".

[31] Q. Zhou, A. Aurelian, C. d. Corral and P. J.Esteban, "A Mciroassembly Station with Controlled Environment," in *Microrobotics and Microassembly,Brandley J.Nelson,Jean-Marc Breguet, Editors,* SPIE, 2001.

[32] P.I.Corke, "Visual Control of Robot Manipulators- A review," in *Visual Servoing*, Singapore, World Scientific, 1993, pp. 1-31.

[33] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *Robotics & Automation Magazine, IEEE,* vol. 13, pp. 82-90, 2006.

[34] F. Chaumette and S. Hutchinson, "Visual servo control, Part II: Advanced approaches," *Robotics and Automation Magazine,IEEE,* vol. 14, pp. 109-118, 2007.

[35] K. K. a. M. Buss, "Multi-Focal Visual Servoing Strategies," in *Vision Systems: Applications, Goro Obinata and Ashish Dutta (Ed.),* I-Tech Education and Publishing, June, 2007, pp. 33-48.

[36] R.Y.Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lens," *IEEE J.Robotics Automat,* no. RA-3, pp. 323-344, 1987.

[37] A. Hornberg, Handbook of Machine Vision, Wiley-VCH, Jan 2007.

[38] A. Z. Richard Hartley, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

[39] "OpenCV v2.4.0-beta2 documentation," 26 04 2012. [Online]. Available: http://opencv.itseez.com/trunk/index.html. [Accessed 27 04 2012].

[40] Z. S.Roth, B. W.Mooring and B. Ravani, "An overview of robot calibration," *IEEE Journal of Robotics and Automation,* vol. 3, no. 5, 1987.

[41] B. Mooring, Z. Roth and M. Deriels, Fundamental of Manipulator Calibration, New York: Wiley, 1991.

[42] N. R. F. CARRIÇO, Calibration of High-precision flexure parallel robots, Lausanne: EPFL, 2007.

[43] A. Kawaji, F. Arai and T. Fukuda, "Calibration for Contact Type of Micro-manipulation," in *IEEE/RSJ Interenational Conference on Intelligent Robots and Systems*, 1999.

[44] S. Fatikow, C. Stolle, C. Diederichs and D. Jasper, "Auto-configuration and Self-calibration in Flexible Robot-based Micro/Nanohandling Cells," in *International Federation of Automatic Control*, Milano, September 2, 2011.

[45] V. Essen, J. Hirvonen, P. Saketi and P. Kallio, " Automated Grasping in Manipulation of Individual Paper Fibers," in *International Conference on Manipulation, Manufacturing and Measurement on the Nanoscale,* Changchun, 2011.

[46] E. Lubrano, Calibration of Ultra-high-precision Robots Operating in an Unsteady Environment, Lausanne: EPFL, 2011.

[47] L. Ljung, System identification: theory for the user, 2nd ed., Upper Saddle River, N.J.: PTR Prentice Hall, 1999.

A.APPENDIX A- Python Script for Chessboard Corner Detection

```
import cv2
import os
import numpy as np

cv2.CV_WINDOW_AUTOSIZE

cv2.cv

def RUCornerPos(path,patternSize):
    dir_list = os.listdir(path)
    cor=[]

    for img_name in dir_list:
            print img_name
            img_b = cv2.imread(path + img_name, 0)
            img=cv2.bitwise_not(img_b)
            print path+img_name


            found,corners = cv2.findChessboardCorners(img, patternSize,
flags=cv2.CALIB_CB_ADAPTIVE_THRESH)

            if found:
                term = ( cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_COUNT,
30, 0.1 )
                cv2.cornerSubPix(img, corners, (5, 5), (-1, -1), term)
                cv2.drawChessboardCorners(img, patternSize, corners,
found)
                nPoint=patternSize[0]*patternSize[1]
                A=np.zeros((nPoint,2))
                for i in range(nPoint):
                    A[i]=corners[i][0]

                sum1=np.sum(A, axis=0)
                print sum1
                virtualPoint=sum1/nPoint
                v=np.zeros((1,2))
                v[0]=virtualPoint
                cor.append(v)

            if not found:
                    print 'chessboard not found'


    return cor
```

B.APPENDIX B- Python Script for Stereo Vision Calibration

```
 def triangulate(P1, P2, x1, x2):

    Xcalc = np.zeros((4, x1.shape[1]))

    for ind in range(0, x1.shape[1]):

        x1_ = x1[:, ind]
        x2_ = x2[:, ind]

        # Constructing A
        A = np.zeros((4, 4));

        A[0,:] = x1_[0]*P1[2,:] - P1[0,:]
        A[1,:] = x1_[1]*P1[2,:] - P1[1,:]
        A[2,:] = x2_[0]*P2[2,:] - P2[0,:]
        A[3,:] = x2_[1]*P2[2,:] - P2[1,:]


        # Solving the linear solution with singular value decomposition
        tmp1, tmp2, V = np.linalg.svd(A)

        # Solving the 3D coordinate from V. In Matlab, it is the last
column;
        # in numpy, it is the last row.
        X = V[-1, :]
        X = X/X[-1]

        Xcalc[:, ind] = X

    return Xcalc

def triangulate2(P1, P2, x1, x2):

    Xcalc = np.zeros((x1.shape[0], 4), dtype = np.float32)

    for ind in range(0, x1.shape[0]):

        x1_ = x1[ind]
        x2_ = x2[ind]

        # Constructing A
        A = np.zeros((4, 4))

        A[0,:] = x1_[0]*P1[2,:] - P1[0,:]
        A[1,:] = x1_[1]*P1[2,:] - P1[1,:]
        A[2,:] = x2_[0]*P2[2,:] - P2[0,:]
        A[3,:] = x2_[1]*P2[2,:] - P2[1,:]


        # Solving the linear solution with singular value decomposition
        tmp1, tmp2, V = np.linalg.svd(A)
```

```
  # Solving the 3D coordinate from V. In Matlab, it is the last column;
        X = V[-1, :]
        X = X/X[-1]

        Xcalc[ind] = X

    return Xcalc

def calculateReprErr(P, Xcalc, x):

    # Reprojection errors
    xrepr = P * np.asmatrix(np.transpose(Xcalc))

    xrepr[0] = xrepr[0] / xrepr[2]
    xrepr[1] = xrepr[1] / xrepr[2]
    xrepr[2] = xrepr[2] / xrepr[2]

    repr_err = np.sqrt(np.power(np.transpose(x) - xrepr, 2))

    avg_repr_err = np.mean(np.sqrt(np.power(repr_err[0], 2) +
np.power(repr_err[1], 2)))
    return avg_repr_err
```