A Calibration Method for MEMS Inertial Sensors Based on Optical Techniques

DONG, Zhuxin

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Automation and Computer-Aided Engineering

February, 2008

Thesis/Assessment Committee

Professor Chung, Chi-kit Ronald (Chair)

Professor Li, Wen Jung (Thesis Supervisor)

Professor Liu, Yunhui (Committee Member)

Professor James Mills (External Examiner)

A Calibration Method for MEMS Inertial Sensors Based on Optical Tracking Techniques

submitted by

DONG Zhuxin

for the degree of Master of Philosophy
at The Chinese University of Hong Kong

Abstract

A MAG-μIMU which is based on MEMS accelerometers, gyroscopes and magnetometers has been developed for real-time estimation of human hand motions. Appropriate filtering, transformation and sensor fusion techniques are combined in the Ubiquitous Digital Writing Instrument (UDWI) to record handwriting on any surface. However, because of the sensors' intrinsic bias and random noise such as circuit thermal noise, a calibration system that provides good reference measurement parameters must be used to compare with the output of the MAG-μIMU sensors. In this thesis a novel idea is proposed to calibrate three-dimensional linear accelerations, angular velocities and space attitude through optical tracking techniques. The Optical Tracking System (OTS) developed by our group consists of two parts: 1) 2D Motion Calibration that is used to obtain linear accelerations of the UDWI in a particular frame defined by us; and 2) the Multiple Camera Calibration that is used for attitude calibration of UDWI. Also, an essential relationship to transform reference frames and angular velocities can be guaranteed after real-time attitude calibration. Hence, the entire nine-dimensional output of the MAG-μIMU can be rectified according to the more accurate data obtained from the optical techniques.

一種基於光學跟蹤技術的 MEMS 慣性感測器的標定方法

董竹新

香港中文大學
自動化與電腦輔助工程學課程
哲學碩士論文

2007年8月

摘要

一個集成磁力計以及MEMS加速度計、陀螺儀的捷連式微型慣性測量單元（MAG-μIMU）的設計方案已被開發出，應用於實時測量、記錄三維運動信息。通過多感測器融合技術，無源慣性制導技術，數字濾波演算法，基於此捷連式微型慣性測量單元的新型數碼書寫系統（Ubiquitous Digital Writing Instrument）能夠實時地跟蹤、記錄三維運動筆跡，而無需額外感測器的輔助。但由於感測器本身的測量偏差，一個可提供足夠好的測量參數的標定系統需要被應用於與感測器輸出的比較當中。本文提出了一種新穎的通過光學跟蹤技術（Optical Tracking Techniques）來標定三維加速度計、陀螺儀及空間姿態的方法。此光學跟蹤系統可分爲兩部分。首先是平面運動的標定,當新型數碼書寫系統工作時通過它可獲得相關加速度；另一個是應用多目攝像機標定（Multiple Camera Calibration）來完成新型數碼書寫系統的姿態標定。此外，借助即時姿態標定還可以得出一種必不可少的用來轉化相關空間結構的關係和角速度資訊。這樣一來, 根據來自光學跟蹤技術的更爲準確的資料，捷連式微型慣性測量單元全部的九維輸出都可以得到改正。

Acknowledgements

This thesis would not have been completed without the help of many people.

I would like to express first and foremost my gratitude to my Master Degree supervisor, Professor Li Wen Jung, for his tireless guidance in the past two years. He has provided creative ideas for my research work and allowed me to participate in the ITF Digital Writing Instrument System (DWIS) project.

I would like to thank Professor Leong Heng Wai Philip who is the PI of the ITF Digital Writing Instrument System project. A number of discussions with him gave me valuable suggestions and comments for improving this work.

I also want to thank Dr Zhang Guang Lie, who has been the team manager of the DWIS project. I have learned so much expertise in many fields and benefited from his selfless and inspiring advice. The time that we spent building the instrument from the ground up, part by part, code by code, will always be remembered by me in years to come.

I would like to give special thanks to my team members, Mr Shi Guang Yi for prototype development, Mr Tsang Chi Chiu for software design, Mr Luo Yi Lun for testing and experiment, Miss Kwok Sze Yin for hardware packaging design, and Miss Wong Yee Yan for marketing analysis, and also for their great teamwork and invaluable discussions.

I am also very grateful to my colleagues, Mr Chung Chor Fung, Mr Chow Chun Tak, Miss Mandy Sin, Mr Chan Cheung Shing, Miss Chow Wing Yin, Mr Xiao Peng, Miss Amy Hiew, Miss Ouyang Meng Xing and Dr Qu Yan Li, for their support to my research and bringing me a comfortable working atmosphere.

Last but not least, I would like to thank my family, my friends in Hong Kong, and especially my parents for all their trust, love, and care.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the non-stop progress of science and technology, especially in the category of micro-electro-mechanical systems (MEMS), it is now possible to fabricate cheap single chip accelerometers, gyroscopes or other sensors, which have already been used in many applications. Inertial measurement units (IMU) in combination with other techniques, are becoming more and more popular in the fields of navigation and motion tracking applications. Because of the steadily decreasing cost and size as well as the improved performance of MEMS inertial sensors, a Micro IMU (μIMU) can also be used for low cost applications, such as orientation devices and human tracking applications [1], [2].

In this project, our overall goal is to develop a system to track handwriting motions without the need of wave sources such as ultrasonic or infrared signals. In sourceless navigation technology, such as inertial kinematics theory, accurate attitude is fundamental to determine and to keep track of the rigid-body position in space. Due to nonlinearity in the system dynamic equations, bias error and random walk noise from attitude sensors will be accumulated and magnified, leading to nonlinear distortions in position tracking.

Attitude tracking is widely used in navigation, robotics, and virtual reality. Classically, the problem of distortions in position tracking is addressed by the Attitude Heading Reference System (AHRS) [3], [4]. The AHRS utilizes gyroscope propagation for transient updates and correction by reference field sensors. However, classically, the performance is ensured by extremely accurate sensors and hardware filters. Due to its expensive cost and large system size, the AHRS has been limited in applications, especially for mobile human position tracking applications.

With MEMS sensing technology, the inertial sensors could be built with low cost and small sizes. However, they suffer in accuracy when compared with bulkier sensors. Nevertheless, new reliable attitude tracking systems have been developed based on low cost gyroscope sensors and the Global Positioning System (GPS). For feedback correction, Euler angles are derived from GPS to represent spatial rotation and a Kalman filter is implemented to fuse with the attitude propagation. But GPS signals are not available for indoor applications and the GPS attitude has resolution limit for handwriting application [5].

For static applications such as the Unmanned Ground Vehicles (UGV) control [6], the MEMS accelerometers work reliable as gravity sensors. Euler angles can be derived directly [7]. However, during a dynamical situation, the accelerometer measurements for the gravitational accelerations will be interfered with the inertial accelerations, which then cannot be trusted for attitude. Further, the pitch attitude along the gravity axis cannot be determined.

Magnetometers experience no such crosstalk disturbance in both situations. However, following the same approach, attitude ambiguity occurs along the magnetic field direction and Euler angles cannot be derived directly. Furthermore, the Earth magnetic field is overlapped by random noise from electromagnetic interference (EMI).

A Ubiquitous Digital Writing Instrument (UDWI) has been developed by our group to capture and record human handwriting or drawing motions in real time based on a MEMS- μIMU [8]. Although both the hardware and software of the UDWI have been steadily improved over the past few years [9], [10], noise signal can still occur and affect the UDWI system output. The noise may include the intrinsic drift of the sensors, misalignment of the sensors during PCB integration, random noise and so on, which are impossible to eliminate completely. Then, in reality, we have to compensate the sensor drift after a hand-writing stroke is completed (see Figure 1.1), which would lead to a delay during real-time hand-writing recognition. However, if a model is available for real-time errors in advance, a more effective compensation algorithm could be developed to overcome the drift and noise.

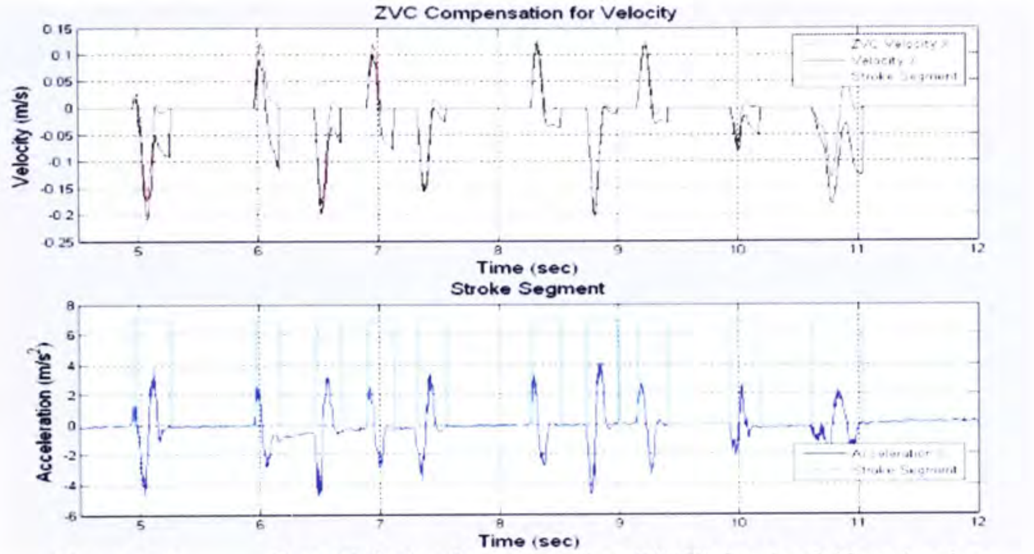


Figure 1.1 Zero Velocity Compensation (ZVC) for the drift and noise

1.1 Architecture of UDWI

Figure1.2 illustrates the block diagram of a wireless MAG-μIMU with the real-time attitude filter system.

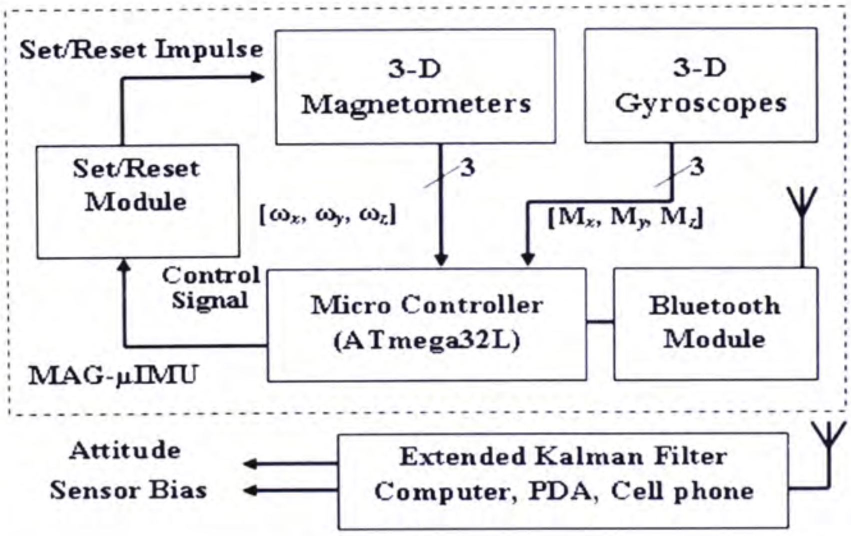


Figure1.2: Wireless MAG-μIMU block diagram

The MAG-μIMU is developed for a wireless digital writing instrument and used to record human handwriting. The μIMU integrates 3D accelerometers [11] and 3D gyroscopes [12] with strap-down installation [8]. The 3D magnetometers, 'MAG' sensors, are added to measure the Earth magnetic field [13], [14], [15]. The sensor unit is fixed on to a commercially available marker to measure the inertial and magnetic information in the pen's body frame.

The output signals of the accelerometers [A_X, A_Y, A_Z] and the gyroscopes [ω_X, ω_Y, ω_Z], which are the body frame accelerations and the roll, pitch, yaw angular rates, respectively, will be measured directly through an Atmega32L A/D converter microcontroller [16], [17]. The serial Bluetooth transceiver is implemented via a USART connection with the MCU for wireless communications [18], [19]. The serial USB transceiver is integrated for transfer backup and hardware development [20]. The digital sample rate of the sensor unit is 200 Hz and the transmit baud rate is 57.6 Kbps via Bluetooth wireless connection, which ensures rapid response to human handwriting.

Figure 1.3 shows the MAG-μIMU version 1.1 with strap-down gyroscopes and magnetometers for attitude tracking test. The sensor system utilizes four-layer printed circuit board techniques for noise reduction. The dimensions are within 56×23×15mm. Figure 1.4 shows the MAG-μIMU version 1.2 combining accelerometers for the handwriting experiment. The sampling rate for nine-channel inertial and attitude sensors is 200 Hz. The dimensions of the sensor unit are within 65×24×15mm. Figure 1.5 illustrates the MAG-μIMU sensor structure of the digital writing system for position tracking.

Based on the strap-down kinematics theory [21], the body frame accelerations can be transformed to the Earth frame through a Direct Cosine Matrix (DCM). Then the translated accelerations can integrate into 3D trajectories after compensating for the gravitational and rotational ones. Finally, any 2D human handwriting is recorded in real time if the pen touches a white board plane. In our design, the accelerometers are mounted as close as possible to the pen-tip for more sensitivity to handwriting motion and approached by the rotational accelerations while the magnetometers are fixed to the pen bottom to reduce magnetic field distortion effect by the metal white board.

1.2 Background of IMU Sensor Calibration

Generally speaking, there are three essential elements to carry on the work of calibration: at first, put the sensors in a controlled environment; then, generate physical reference signals for sensor excitation, and measure the sensor output signal and finally correct the signal transfer based on those calibration measurements. Previously, the calibration of an IMU could be done through a mechanical platform, turning the IMU into different precisely controlled orientations and at known rotational velocities [22], [23], [24]. At each orientation the output of the accelerometers and gyros during the rotations are observed and compared with the pre-calculated gravity force and rotational velocity, respectively. However, the cost of the mechanical platform can exceed by many times the cost of developing and constructing a MEMS sensor-based inertial measurement unit. Furthermore, although the load capacity is not a problem (~80 grams), current motorized motion stages do not have enough degrees of freedom to emulate the motion of our hands in spite of being able to provide the higher accuracy. Finally, most of these motion stages can only implement small accelerations while the majority of our hand motion is made up of much larger accelerations. Generally, the accelerometers will not respond to these 'quiet' motions. Figure 1.6 shows the mechanical platforms we have used before which include an X-Y table to provide translation and a rotation stage. These platforms were not sufficient for MEMS motion sensor calibration due to their mechanical vibrations. In addition, there are also other equipments for the calibration in the market. For example, the two machines shown in Figure 1.7 are the products for The Model Shop INC. [25]. Owing to these probable drawbacks, some other methods have been developed [26]. Enlightened by these considerations and according to our current situation, a helpful calibration method has been developed to service the UDWI.

Many matching algorithms and techniques have already been developed and used in the category of motion estimation with the aid of video compression. For example, there are already Three Step Search (TSS) [27], [28], Four Step Search (FSS) [29], and Parallel Full Search (PFS) [30] in existence and being used to carry out this kind of experiment.

Also, there exist several kinds of matching criteria, e.g., Correlation Coefficient (CC), Block Distortion Measure (BDM) [1] and Mean Absolute Difference (MAD) [30]. All of these methods are matured and reliable to implement the estimation of motion vector through video sequences. In our experiment, we adopted an improved search method, which is based on the PFS, together with CC as the criteria to make up a matching algorithm. PFS is a widely used basic method, which is most stable and convenient for us to realize. One of the advantages is that it can save time significantly since we can define the search area freely according to the template definition and that makes this algorithm very effective. Another advantage is the higher accuracy of seeking the template in the next pin image because of the smaller but closer area in which we search for it. Obviously, for some very small motions, which will probably happen frequently because of the high sampling rate we set, there must be measurement errors to affect the reconstruction of characters owing to the limited resolution of the matching program.

Besides the process of matching, attitude calibration is also necessary for the entire MAG-μIMU system calibration. Here, we need a multiple camera calibration system to reconstruct a line which can represent the attitude of the digital pen in a particular 3D frame. A Camera Calibration Toolbox for MATLAB [31] is our fundamental tool in this part. This toolbox can provide us with both single camera calibration through corners extraction and the calibration of the stereo camera system. We can obtain such helpful parameters as camera intrinsic parameters, rotation matrix and translation vector. Based on these parameters some other functions have been developed and added into the toolbox, which can help us reconstruct a 3D point described in this particular frame from the information of its 2D image coordinates. When a line can represent the gesture of the UDWI and at the same time any two points on it are reconstructed, the attitude can be consequently calculated out. Hence, a system for attitude calibration can be developed including the information of both positions and angles.

Another important experimental device for sensor calibration is a High Speed Camera [32], which plays an important role in both 2D motion estimation and 3D attitude calibration. The camera is available with various image sensors. Depending on the sensor selected, the camera generates digital black-&-white or color images with various spatial resolutions and exposure times.

1.3 Organization

The remaining chapters of the dissertation are organized as follows. In Chapter 2, the design of the 2D motion calibration system is introduced, including the set-up of the experimental platform and software algorithm. Chapter 3 and Chapter 4 describe the camera calibration and multiple camera calibration system we developed. Based on the 2D motion estimation and the 3D reconstruction through the camera calibration systems, data can be obtained from the calibration system considered as a reference for the inertial sensors and magnetometers. Experimental results will be illustrated and discussed in Chapter 5. Chapter 6 gives another experimental set that includes a single letter with multiple tests. Then, based on these given information, a potential method for letter classification is introduced, including the experimental schematics and results. Finally, the conclusions and proposed future improvements are stated in Chapter 7.

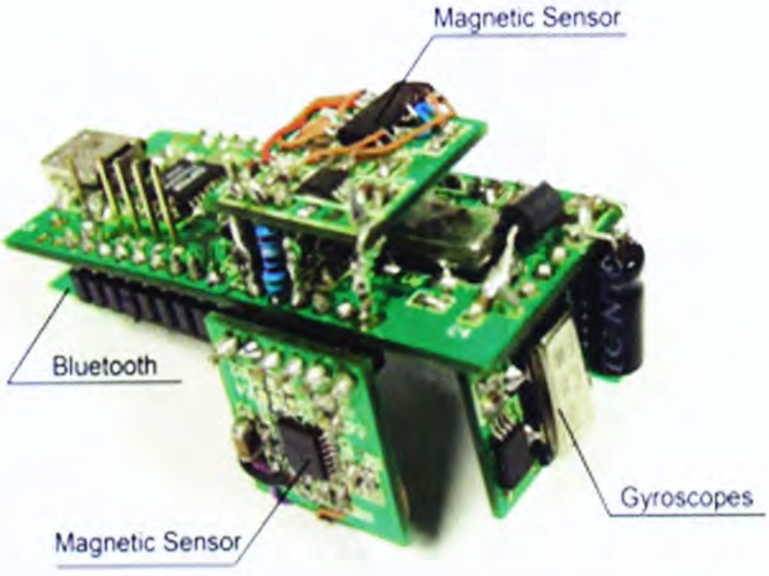


Figure 1.3 The Prototype of the MAG-μIMU version 1.1 with Bluetooth Module

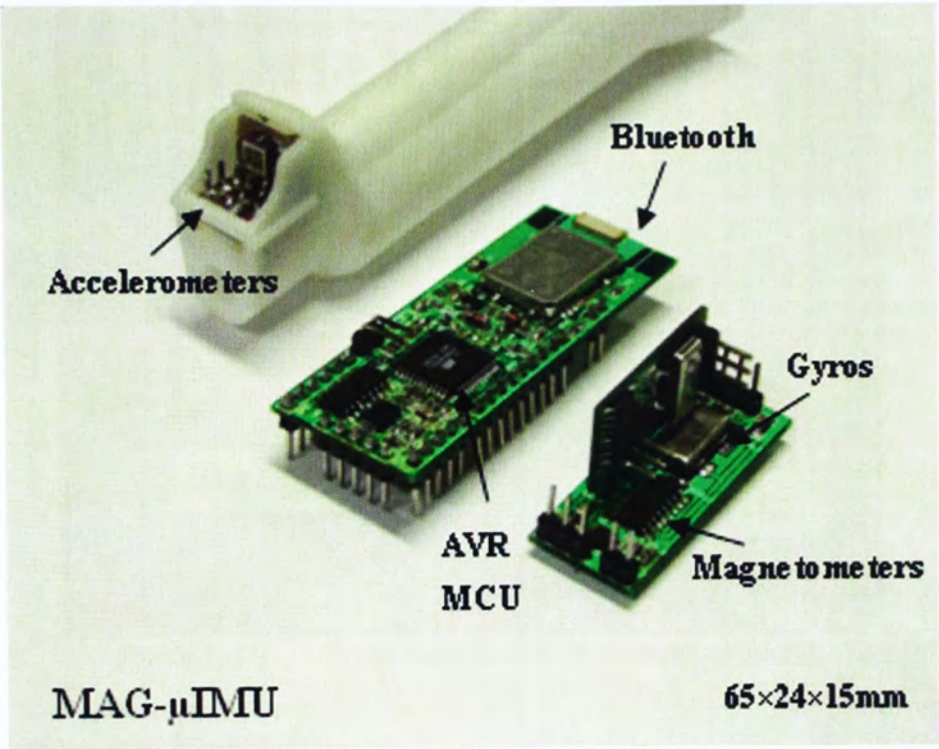


Figure 1.4 MAG-μIMU version 1.2 for Digital Writing Instrument application

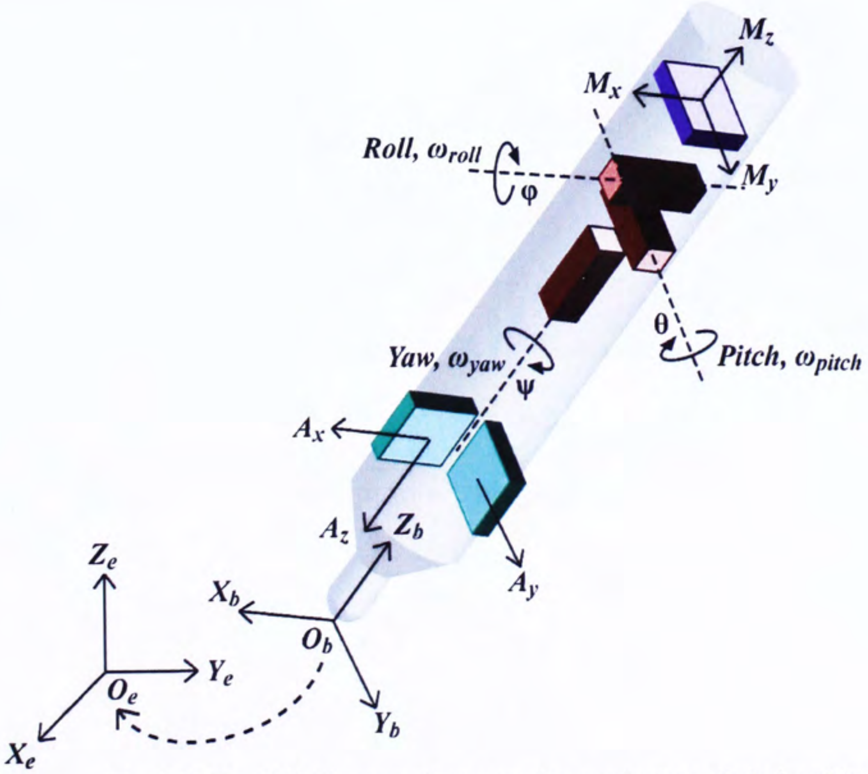


Figure 1.5 The MAG-μIMU System Structure for a wireless digital writing instrument

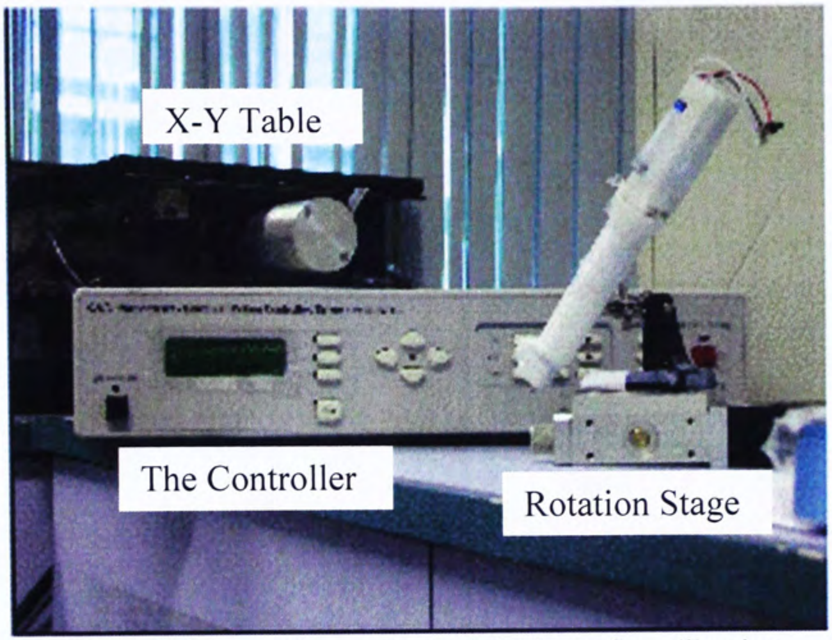
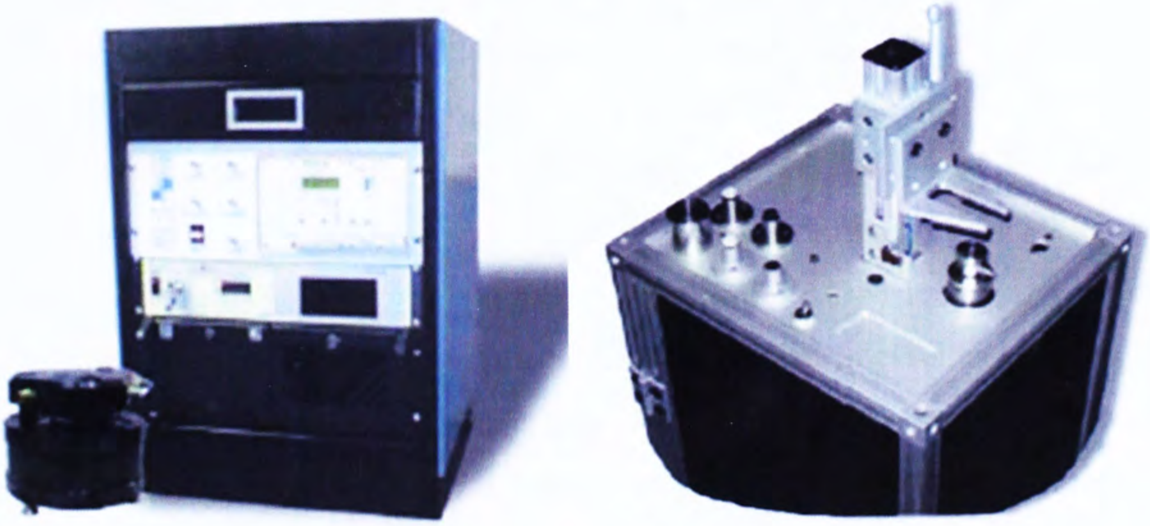


Figure 1.6 The mechanical platforms for calibration



(a) acceleration calibration workstation (b) shock exciter

Figure 1.7 Examples of existing calibration machines

Chapter 2

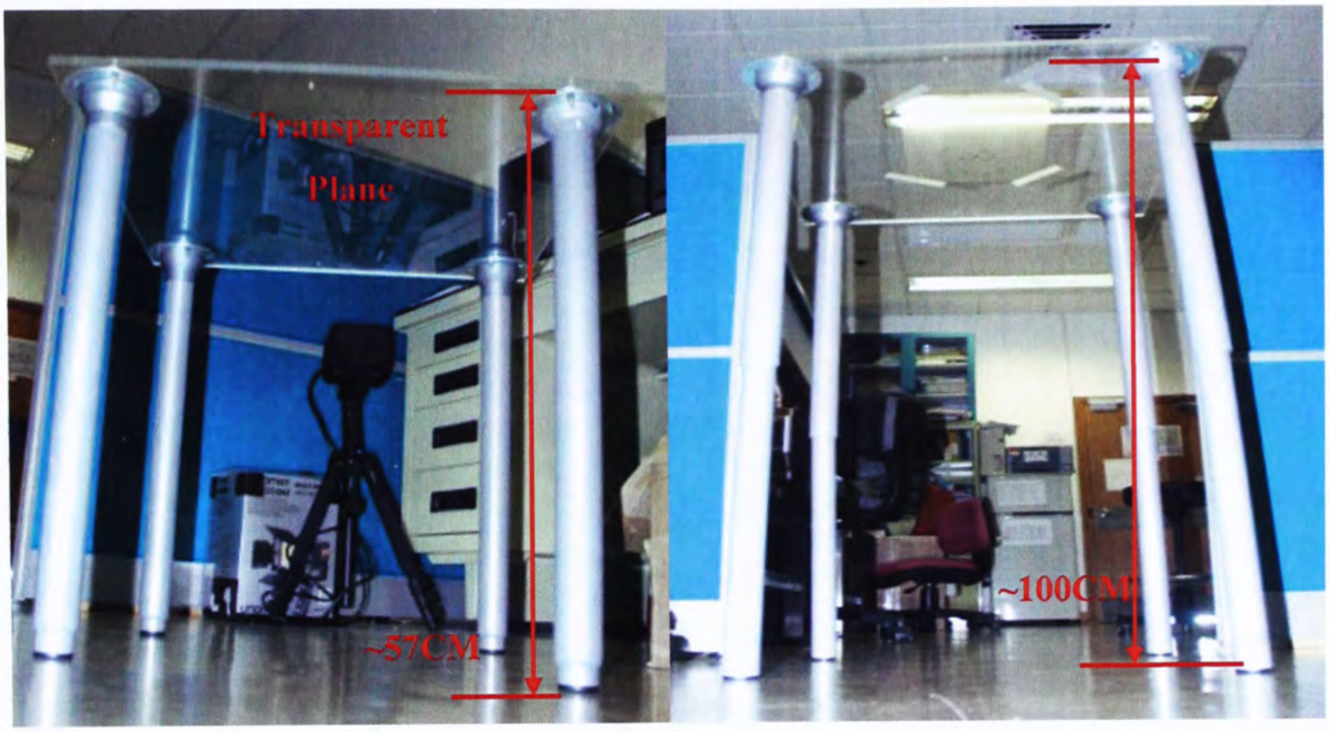
2D Motion Calibration

2.1 Experimental Platform

The concept is that, in a recorded video, the motion of a pen with μIMU sensors installed is registered while it is writing on a marked paper with a fixed gesture. The video can then be divided into sequential frames, and the frames are analyzed to calculate the position of any particular visible point in the pictures, so that the motion track of the pen can be detected from these frames. Since the sampling frequencies of our IMU sensors are set at 200Hz, so a high speed camera is needed to synchronize the video data to the sensors' output. The trajectory of the pen on the paper can be recorded with markers, from which the displacement of the pen-tip can be read out in pixel resolution. With the help of the markers, the length of which is known in advance, it is convenient to calculate the proportionality coefficient between pixel and physical unit. Hence, motion vectors can be estimated for the velocity and acceleration calculation of the pen-tip, which will be used to calibrate the IMU sensor since its motion is the same.

2.1.1 Transparent Table

In order to obtain a clear view of the whole motion of the pen-tip while writing, the camera is located directly below the pen, which will avoid the intrusive effects of the hand or something else in view. Obviously, the table plane is required to be transparent to observe the actions. Accordingly, we constructed the table illustrated in Figure 2.1.



(a) table in common state

(b) table after length adjusted

Figure 2.1 The transparent table with adjustable height

This table has telescopic legs, which give it a maximum height of 1000mm. Owing to both the transparent plane and the adjustable legs, it is feasible for us to build a 2D calibration system as shown in Figure 2.2 and Figure 2.3.

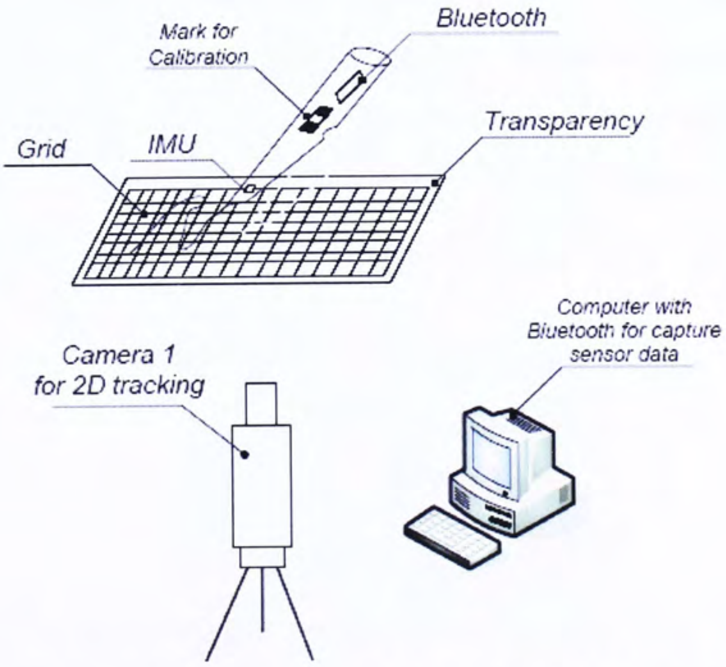


Figure 2.2 Schematic of 2D motion calibration

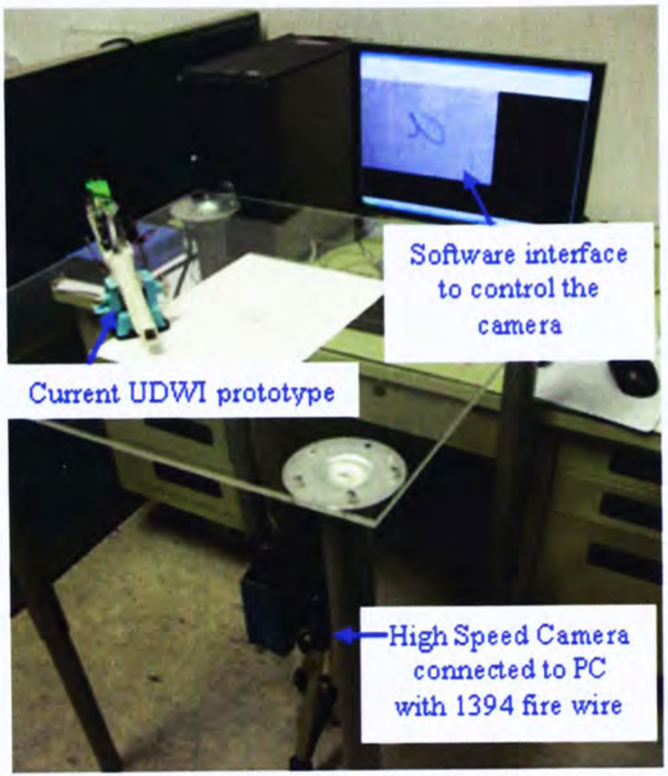

Software interface to control the camera
Current UDWI prototype
High Speed Camera connected to PC with 1394 fire wire

Figure 2.3 Experimental platform of 2D motion calibration

In addition to this table, a transparent 'paper' is also necessary to carry out the experiments. The transparency has markers on it as shown in Figure 2.4, which brings two advantages to our experiments. One is that the length of the markers is a constant, 16mm here. In this case, when we obtain the pixel information of the two pole points of the markers, the proportional parameter between pixel and millimeter can be obtained. The other is to overcome a drawback of the camera operation system, because when it is ready to record the video, you cannot see the real-time view from the camera operation window. So it is very helpful to define a writing area which can be observed completely. The area bordered by the markers is a rectangular shape of 71mm × 50mm.

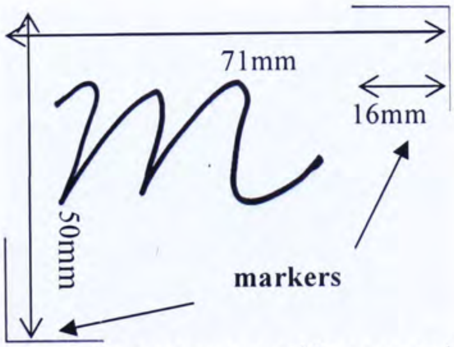


Figure 2.4 A transparency with paper as background

When something is written on the table, with this set up, a clear observation can be obtained through the image sequence recorded by this system. Figure 2.5 presents one picture of the image sequence while writing the letter *a* on the table, recorded by the high speed camera. In this way, the whole course of writing can be represented by the sequence, which is very useful to analyze the motion.

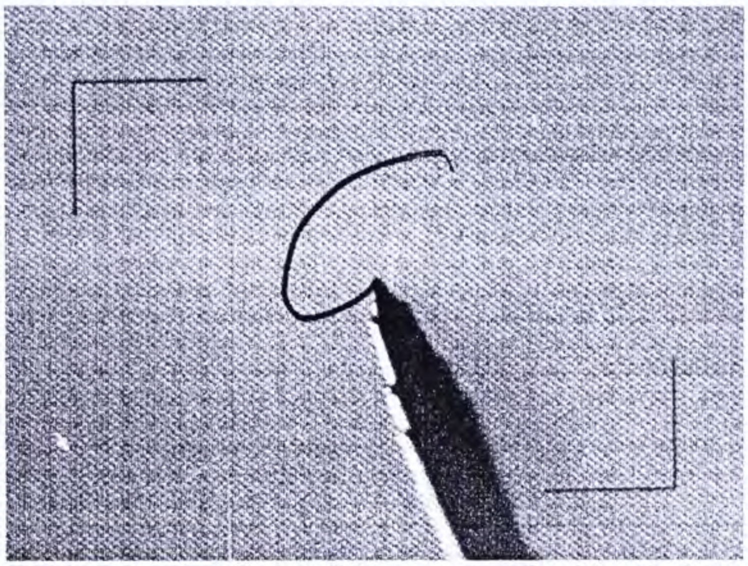


Figure 2.5 One image of the sequence of writing the letter “a”

2.2 Matching Algorithm

2.2.1 Motion Analysis

It is a key-point for us to figure out how the pen moves while writing, especially the pen-tip, since the accelerometer are situated there. In other words, the accelerations are going to be calculated out on this 2D plane and finally they will be compared with the output of the accelerometers as the reference to calibrate them. The clue is the various location of the pen-tip during the course of its motion, through which the motion vector can be estimated. Figure 2.6 shows how we estimate the motion vector between two neighboring pictures. As the time interval is known in advance, the velocities and accelerations can be obtained as well. With this method, the estimation is repeated one by one for each pin along the entire sequence. Then a body of data, which can describe the motion properties, is achieved.

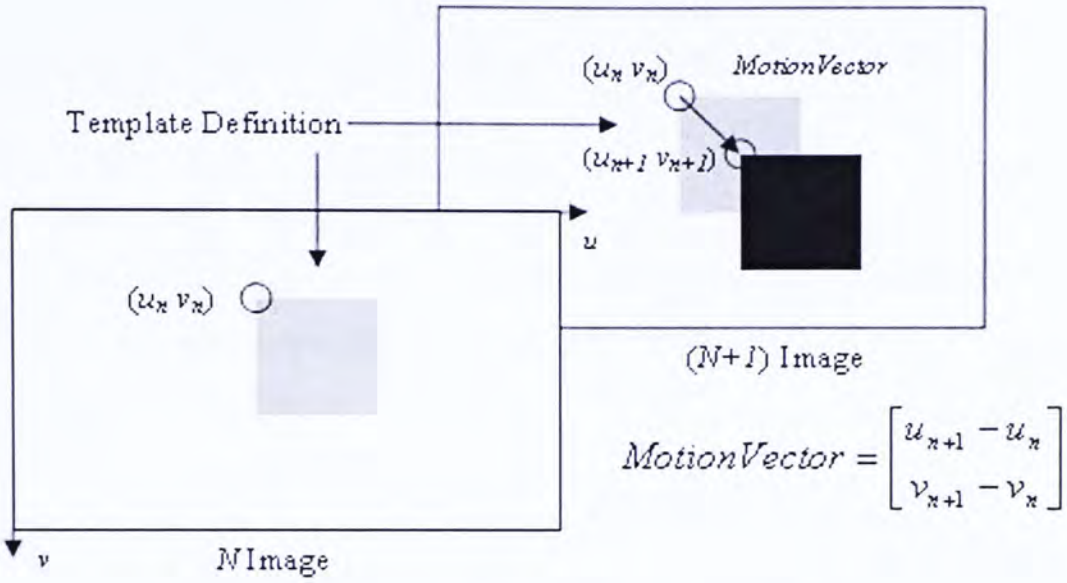


Figure 2.6 Estimation of motion vector

2.2.2 Core Algorithm and Matching Criterion

When we make use of the function of the white-&-black optical sensor of the camera, a sequence of gray-scale images can be obtained and every pixel can be distinguished from its gray value, ranged from 0 to 255. Hence, obviously, the different areas with various bright and dark properties can be recognized. According to the Correlation Coefficient theory, an algorithm of matching is developed.

In the program, firstly, a template will be defined in a rectangular shape enclosing the pen-tip area according to its location in the image (see Figure 2.8). This template is able to be updated one by one during the whole search. For instance, the most similar block to the template defined in the first picture could be found in the second image after matching. Then the program can automatically consider this block as a new template instead of the original one and continue matching in this way till the end. In this way, even if the pen-tip changes a little during writing we can still complete the matching to get a good result. The upper-left corner coordinate will be recorded as the position of the template and the size of it is known as well. Then the calculation of the sum of gray values is presented as follows:

$$dSigmaT = \sum_{n=v}^{THeight+v-1} \sum_{m=u}^{TWidth+u-1} \left(Xt(m,n)\right)^{2} \qquad (1)$$

where *u* and *v* are the coordinates of the upper-left corner; *Xt* is the gray value and *THight* and *TWidth* are the height and width of the template, respectively. *dSigmaT* is the sum of gray values squared of the template. Then the whole picture is searched block by block with the same size as the template.

$$dSigmaS = \sum_{n=j}^{THeight+j-1} \sum_{m=i}^{Twidth+i-1} \left(Xs(m,n)\right)^{2} \qquad (2)$$

where *j* and *i* are the pixel coordinates of the first point. They vary from 0 to (*PWidth-TWidth*) and (*PHeight –THeight*) respectively, where *PWidth* is the width and *PHeight* is the height of the picture. $(PWidth - TWidth + 1) \times (PHeight - THeight + 1)$ of blocks will be calculated all over the picture.

The classical theory of defining a correlation coefficient can be presented as follows:

$$\rho_{xy} = \frac{Cov\left(X, Y\right)}{\sqrt{D\left(X\right)} \cdot \sqrt{D\left(Y\right)}} = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right) \cdot \left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \cdot \sqrt{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}} \qquad (3)$$

where *X*, *Y* are two group variables, *Cov(X,Y)* presents the covariance between *X* and *Y*; *D(X)* and *D(Y)* are the variances respectively of *X* and *Y*. As to the ρ_{xy}, it is supposed to be a value within [0 1]. The higher the ρ_{xy} is, the more similar *X* and *Y* are to each other. In the algorithm, the two variables are the sum of gray values of the pixels in the template and the searched block.

$$dSigmaST = \sum_{n=j}^{THeight+j-1} \sum_{m=i}^{Twidth+i-1} Xt(u+m-i, v+n-j) \cdot Xs(m,n) \qquad (4)$$

where *dSigmaST* is the covariance. The correlation coefficient can thus be calculated as follows:

$$\rho = {dSigmaST}\Big/\left(\sqrt{dSigmaS} \cdot \sqrt{dSigmaT}\right) \qquad (5)$$

Comparisons will be completed among these ρ and the block which has the highest ρ is able to be picked out. We consider this block as the exact place where the pen-tip moves to and we describe the location of the block also with its upper-left corner. Then the estimation about the motion vector can be implemented.

If we keep the relative position and orientation consistent between the camera and table, and also the focus of the camera, the conversion coefficient K between the pixel distance and physical distance will be a constant. It can be calculated out according to the markers. Then we can carry out the calculation of velocities and accelerations by means of the mathematic model as follows:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \cong \frac{K}{\Delta t} \begin{bmatrix} u_{n+1} - u_n \\ v_{n+1} - v_n \\ 0 \end{bmatrix} \qquad (6)$$

$$\begin{bmatrix} acc_x \\ acc_x \\ acc_z \end{bmatrix} \cong \frac{1}{\Delta t} \begin{bmatrix} V_{xn+1} - V_{xn} \\ V_{yn+1} - V_{yn} \\ g \cdot \Delta t \end{bmatrix} \qquad (7)$$

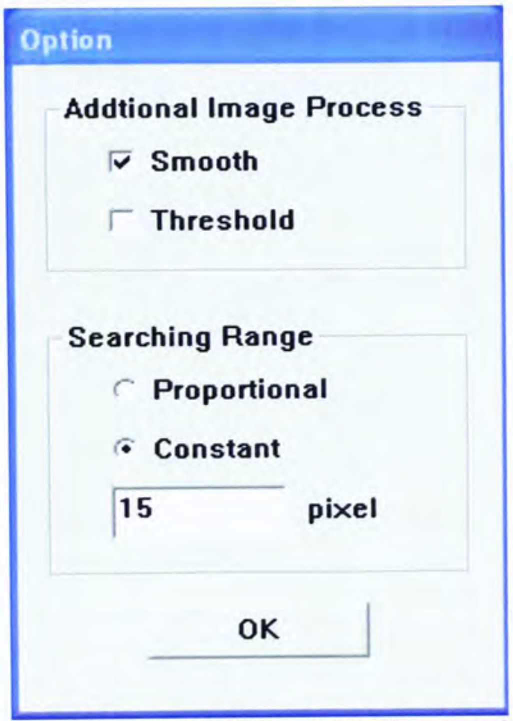
where u_f and v_f are the pixel coordinates results in the final found block (see Figure 2.6); u_s and v_s represent the position of the source block; Δt depends on the sampling frequency of the camera, i.e., $\Delta t = \frac{1}{f}$. Equation (7) above is guaranteed only if the motion plane is absolutely horizontal. Otherwise, the gravity g will take effect on the results of the 2D calculated accelerations.

2.3 Usage of High Speed Camera

The multiple functions of the image sensors in the *pco.camera* system [32] have proved very convenient as we can capture images both in black-&-white and color format easily, but the most important thing is the sampling rate. Since the IMU is set at a rate of 200Hz to capture data, there has to be a picture sequence captured at the same rate to synchronize with it. Fortunately, the camera can reach a sampling frequency of 1000Hz at most. So it can satisfy the experiment when the exposure time is 0.005 second. The *pco.camera* system also includes a digital image output (the 1394 Firewire 400), a separate power supply and image processing and camera control software. Besides the camera, a lens and enhancing light will also be brought into this system, to make sure the experiment is carried out smoothly. More experiments have been performed on the camera system and the lens, details of which will be introduced in the next chapter.

2.4 Functions Realized

Based on the techniques described above, an improvement has been added. Because of the high sampling rate, the movement of the pen tip between two neighboring pin images is not obvious and significant. Hence searching the whole image to find the most similar block proved not that effective and took a lot of extra time. Another few functions have been added into this algorithm, as shown in Figure 2.7. This provides an alternative between proportional and constant models of the searching range. In the former model, an expanding area based on the defined template three times larger will be considered as the destination range; while during the latter, one can input a constant in unit of pixel, then the area enlarged by the constant from the template in the orientations of up, down, left and right will be the target. In addition, the image can also be dealt with using the smooth function and threshold function. The smooth function is the default setting in our current algorithm, which is used to filter some pixels working as a low-pass filter. The gray values of these too bright or dark points will be replaced by that average with the surrounding points.



Option

Addtional Image Process

☑ Smooth

□ Threshold

Searching Range

○ Proportional

● Constant

15 pixel

Figure 2.7 Model option before matching

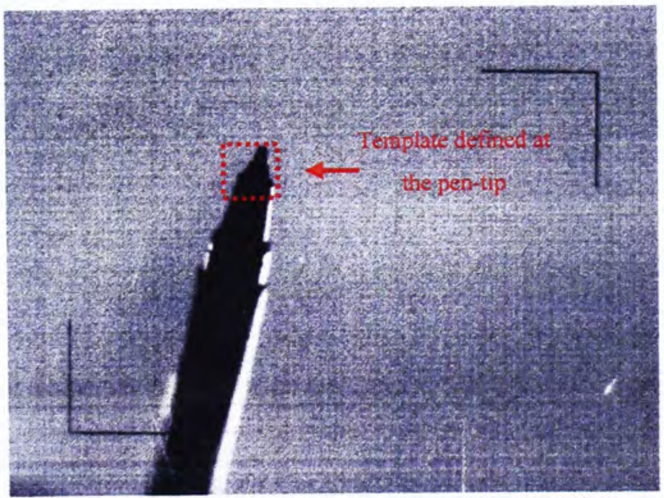
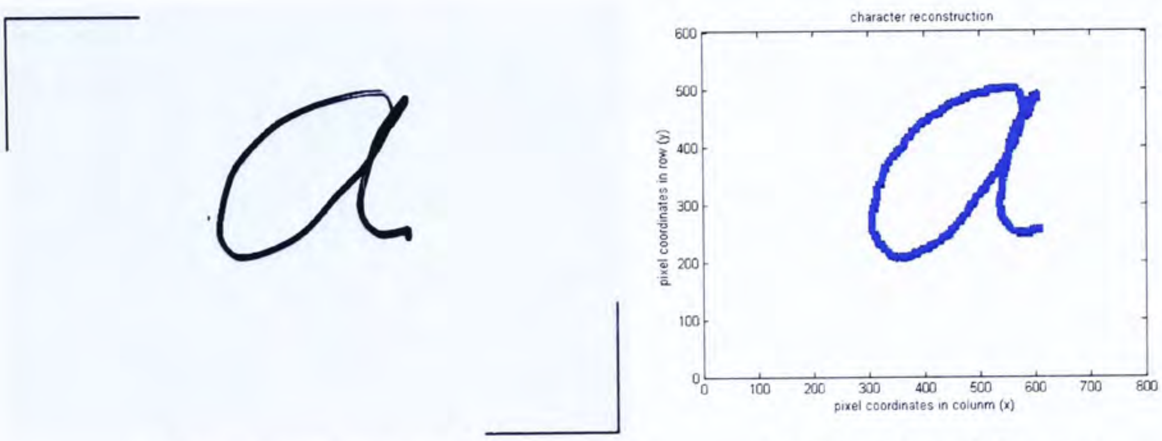


Figure 2.8 The definition of the template

In this test the letter *a* has been written with a real pen. The exposure time was set at 0.005 second, which means 200 hundred pictures can be captured per second. The sequence of writing the letter *a* in this case consists of 280 images from the beginning to the end. The duration to write this character was 1.4 seconds. Hence, through our matching algorithm a 2D matrix was applied to describe the positions of all the found blocks while it was moving on the transparent table to complete a hand-writing motion. Finally, the matrix consists of 280 rows and 2 columns of the pixel coordinates. Based on this matrix of position coordinates, we reconstructed the character *a* and compared it with the written one as shown in Figure 2.9.



The written character

The reconstructed character

Figure 2.9 Comparison between the original letter and the reconstructed one

The reconstructed letter is made up of 280 points and looks very accurate. Then using equations (6) and (7), velocities and accelerations in *x* and *y* axis could be calculated as well. According to the markers, the proportional parameter *K* could also be obtained and it is equal to $1.13e^{-4}$ meters per pixel. Figure 2.10 illustrates the velocities and accelerations in both the *x* and *y* axis with respect to time. The graphs specify the exact velocity or the acceleration at every 0.005 second throughout the motion.

The test has clearly proved that the idea based on matching is feasible to provide a relatively satisfactory result as the reference. The comparison between the synchronized matching results and the output of the sensors will be introduced in a latter chapter.

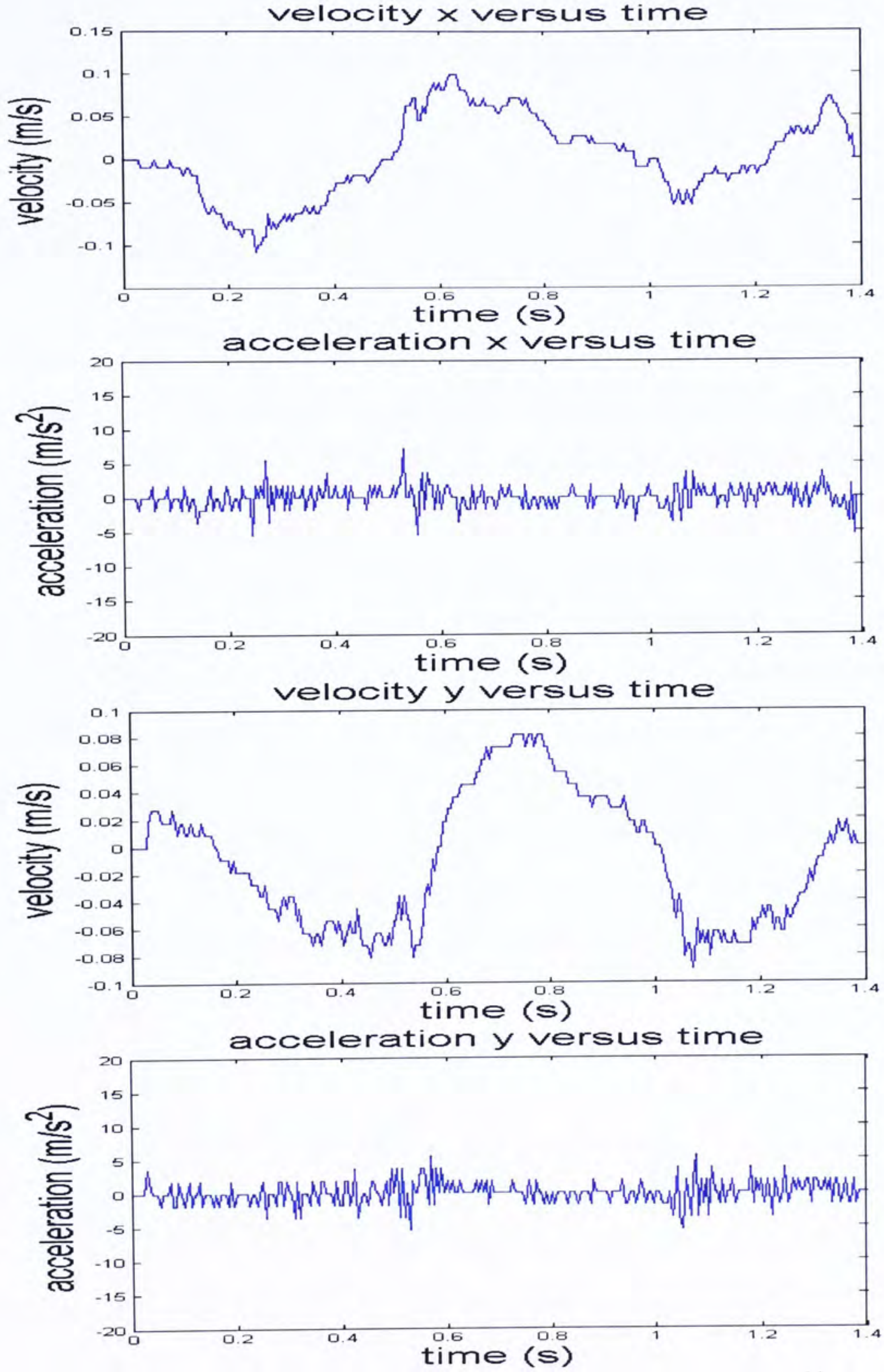


Figure 2.10 Results of velocities and accelerations after matching

Chapter 3

Usage of Camera Calibration

A three-dimensional computer vision system is able to be able to calculate the geometric information of objects in 3D space, such as location, shape and so on, and moreover recognize them according to the information. The brightness of each point in the image presents the strength of the reflected light of the point itself on the object surface while the location of the point in the image depends on the geometric position of another point on the surface of the space object, which correspond to each other. The relationship of these positions is determined by the imaging geometric model of the camera, by which the image is captured. The parameters inside this model are called the parameters of the camera. These parameters can be obtained only by experiments and calculations, and this course is called camera calibration [33]. Camera calibration is the necessary basis of the further goals, such as three-dimensional reconstruction. There will be a detailed introduction on how it works in this chapter.

3.1 Introduction to Camera Calibration

3.1.1 Related Coordinate Frames

In order to describe the optical imaging system clearly, one can define three coordinates. A digital image transformed from the standard signal captured by a camera is stored in the computer as a $M \times N$ matrix. The value of each component in the image of M rows and N columns is the brightness on this image point when the picture is in gray scale format, which, unless specified particularly, we are dealing with in this dissertation. As shown in Figure 3.1, when we define a rectangular coordinate u and v on the image, the coordinate (u,v) of every pixel is its row and column in the matrix. So, (u,v) is the coordinate of the image frame in units of pixels. As this coordinate can only present the row and column without the position in physical unit, another frame should be founded in physical units, for instance millimeters (mm). Some point, O_1, in the image is regarded as the origin of this new image frame while its x and y axes are parallel with the u and v axes respectively. In the coordinate (x,y), the origin O_1 is defined as the intersection of the camera's optical axis and the image plane. In general, it will be located at the center of the image, but somehow there probably could be some variation. Assuming that the coordinate of O_1 in (u,v) frame is (u_0,v_0) and the physical length of each pixel in both x and y axes are dx and dy respectively, then the relationship of any pixel between the two coordinates can be obtained as follows:

$$u = \frac{x}{dx} + u_0 \qquad (8)$$

$$v = \frac{y}{dy} + v_0 \qquad (9)$$

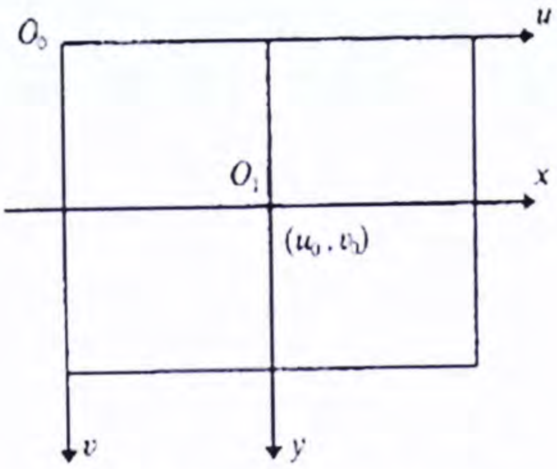


Figure 3.1 2D image frame

Using homogeneous coordinates and in formation of a matrix, the above equations can be rewritten as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{10}$$

The inverse relationship can be calculated as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} dx & 0 & -u_0 dx \\ 0 & dy & -v_0 dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{11}$$

Figure 3.2 illustrates the geometric relationship of the camera imaging system. Inside this system, O is called the optical center of the camera, axis X_c and Y_c is parallel with axis x and y, Z_c is the optical axis, which is vertical to the image plane. The intersection of the optical axis and image plane is the origin of the image frame, O_1. The rectangular coordinate consisting of point O and axes X_c, Y_c and Z_c is named the camera frame while OO_1 is the focal length f of the camera.

Since the camera can be located anywhere at all, a standard frame should be found from the environment to describe the position of the camera and furthermore any object in this environment can also be described through this frame. This frame is called the world frame and is made up of axes X_w, Y_w and Z_w. The relationship between the camera frame and the world frame could be illustrated by a rotation matrix R and a translation vector t. Hence, if the homogeneous coordinates of a space point p in both world frame and camera frame are $(X_w, Y_w, Z_w, 1)^T$ and $(X_c, Y_c, Z_c, 1)^T$ respectively, a relationship is able to be calculated as follows:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M_1 \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{12}$$

where R is 3×3 orthogonal unit matrix; t is a three-dimensional translation vector; $0^T = (0,0,0)$; M_1 is a 4×4 matrix.

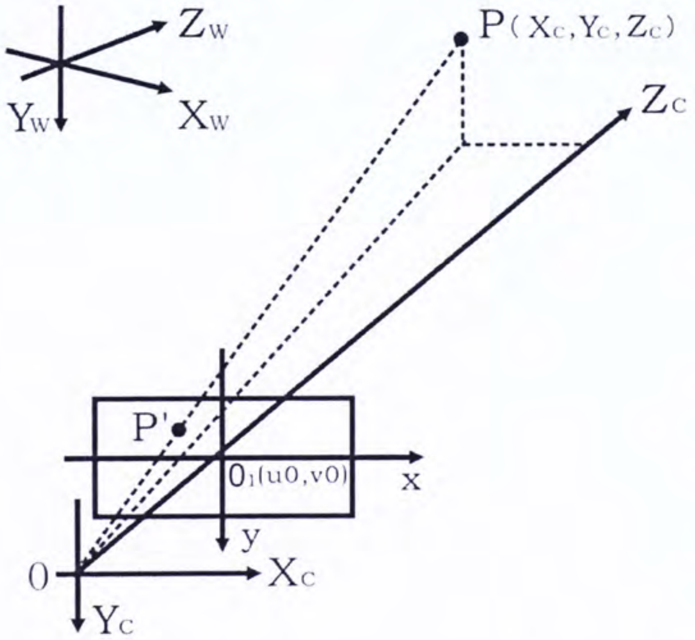


Figure 3.2 The camera frame and the world frame

3.1.2 Pin-Hole Model

The imaging position of any space point P in the image can be almost described by the pin-hole model, which means that the projection position p in the image of point P is the intersection of the line OP and the image plane. This relationship is called the center projection or perspective projection and can be presented as the equations below:

$$x = \frac{fX_c}{Z_c} \tag{13}$$

$$y = \frac{fY_c}{Z_c} \tag{14}$$

where (x, y) is the image coordinates of p and (X_c, Y_c, Z_c) is the coordinates of P in the camera frame. Let us represent this relationship in the form of homogeneous coordinates and matrix:

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{15}$$

$$\begin{aligned} Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} a_x & 0 & u_0 & 0 \\ 0 & a_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M_1 M_2 Xw = MXw \end{aligned} \tag{16}$$

When we submit (11) and (12) into (15), a relationship can be calculated between the coordinates $(X_w, Y_w, Z_w)$of P described in the world frame and the coordinates (u, v) of p as shown in equation (16), where $ax = {}^{f}\!/_{dx}$, $ay = {}^{f}\!/_{dy}$; M is a 3×4 matrix, which is called the projection matrix. As M_1 is determined only by the parameters a_x, a_y, u_0, v_0 which depend on the camera's internal structure, we call these parameters the intrinsic parameters of the camera; while M_2 totally depends on the relative location and orientation, and we call these parameters the extrinsic parameters of the camera. The process of calculating all these parameters including both the intrinsic and extrinsic ones is called camera calibration.

According to equation (16), if the projection matrix is known, which means the intrinsic and extrinsic parameters are obtained, and if the coordinates $Xw = (X_w, Y_w, Z_w, 1)^T$ of any space point P are known too, the coordinates (u, v) of its image point p is able to be calculated. This is because (16) provides three independent equations, based on which it is enough to calculate after Z_c is cancelled. On the other hand, if the coordinates (u, v) of image point p are known, even though the matrix M is also known, the coordinates Xw cannot be relied upon to be specified as the only solution. In fact, M is an irreversible matrix, so when M and (u, v) are known, from equation (16) only two linear equations with respect to X_w, Y_w, Z_w can be guaranteed after Z_c is cancelled. These two linear equations make a radial line of function OP , which means all points which have the same projection point p are on this radial line. In Figure 3.2, when image point p is known, through the pin-hole model, any point lying on OP has the same image point p . In this case, we cannot reconstruct a point from 3D space through camera calibration directly.

3.2 Calibration for Nonlinear Model

As illustrated by experiments, a linear model cannot perfectly describe the imaging geometric relationship, especially when a wide-angle lens is used. There will probably be an obvious distortion at the edge of the image. The further from the image center the worse it will be. Some equations are used to describe the non-linear distortion as follows [34], [35]:

$$\bar{x} = x + \delta_x(x, y) \quad (17)$$

$$\bar{y} = y + \delta_y(x, y) \quad (18)$$

where $(\bar{x}, \bar{y})$ is the ideal coordinates calculated by the pin-hole model; (x, y) is the real coordinates; δ_x and δ_y are non-linear distortion values and related to their locations in the image.

$$\delta_x(x, y) = k_1 x(x^2 + y^2) + (p_1(3x^2 + y^2) + 2p_2 xy) + s_1(x^2 + y^2) \quad (19)$$

$$\delta_y(x, y) = k_2 y(x^2 + y^2) + (p_2(3x^2 + y^2) + 2p_1 xy) + s_2(x^2 + y^2) \quad (20)$$

where the first component of δ_x or δ_y is called radial distortion; the second is decentering distortion; the third is called thin prism distortion; $k_1, k_2, p_1, p_2, s_1, s_2$ are the non-linear distortion parameters.

Generally, the first component is enough to describe the non-linear distortion. In the non-linear optimal algorithms, excessive non-linear distortion parameters could not only improve the accuracy but also lead to unstable solutions [36]. But referring to the wide-angle lens, it will obviously be helpful for more accurate calibration [35], [37].

3.3 Implementation of Process to Calibrate Camera

A toolbox based on MATLAB to carry out camera calibration has been developed and available [31]. Through this toolbox, we can carry out experiments on camera calibration freely and derive more functions to satisfy our own requirements. In this section, I will introduce in detail how we calibrate the camera.

3.3.1 Image Capture

The main object we use the camera to capture images is a 2D plane with a grid of a fixed size on it, as shown in Figure 3.3. After the template is fixed on a flat plane, we use a camera to take a number of pictures of this template from various distances and orientations. We then read these pictures into the calibration system. Hence, all the pictures are ready to calibrate (see Figure 3.4).

3.3.2 Define World Frame and Extract Corners

To extract the corners on the image individually we need to click out the four extreme points on the template, and tell the system from where we are going to extract corners. Another function of this is to define the world frame based on the template plane. Figure 3.5 illustrates the course of defining a world frame. When the length of the grid is input into the system manually, the corners extraction will be finished automatically.

The first click decides the origin of the world frame. This means the point where you click first is regarded as the origin. Then the other three extreme points can be clicked out in any order. The boundary of the calibration grid is determined as shown in the fourth image of Figure 3.5. Make sure that the Z axis points up and according to the right-hand rule, we can specify the world frame with only one case described in green lines. Furthermore, input of the length of the grids after the frame is achieved, for instance 30mm here. Finally, the corners are extracted with the coordinates known both in the image frame and the world frame. If the predicted corners are close to the real image corners are close enough to the real ones. Repeat this operation for each picture of the group, and the extraction of corners is completed.

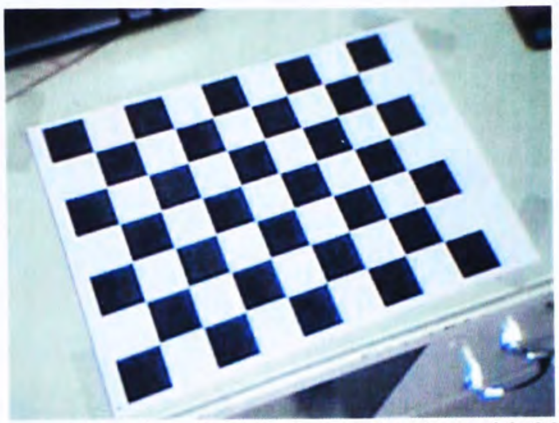


Figure 3.3 Template for camera calibration with 30x30mm grid

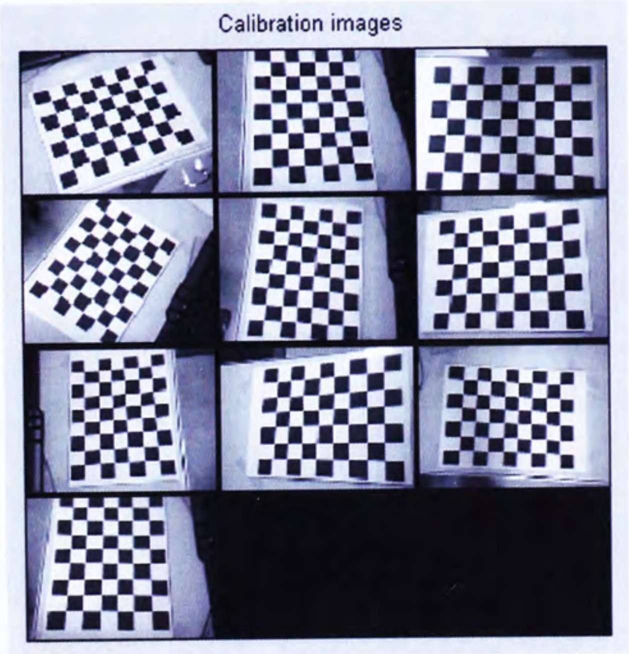
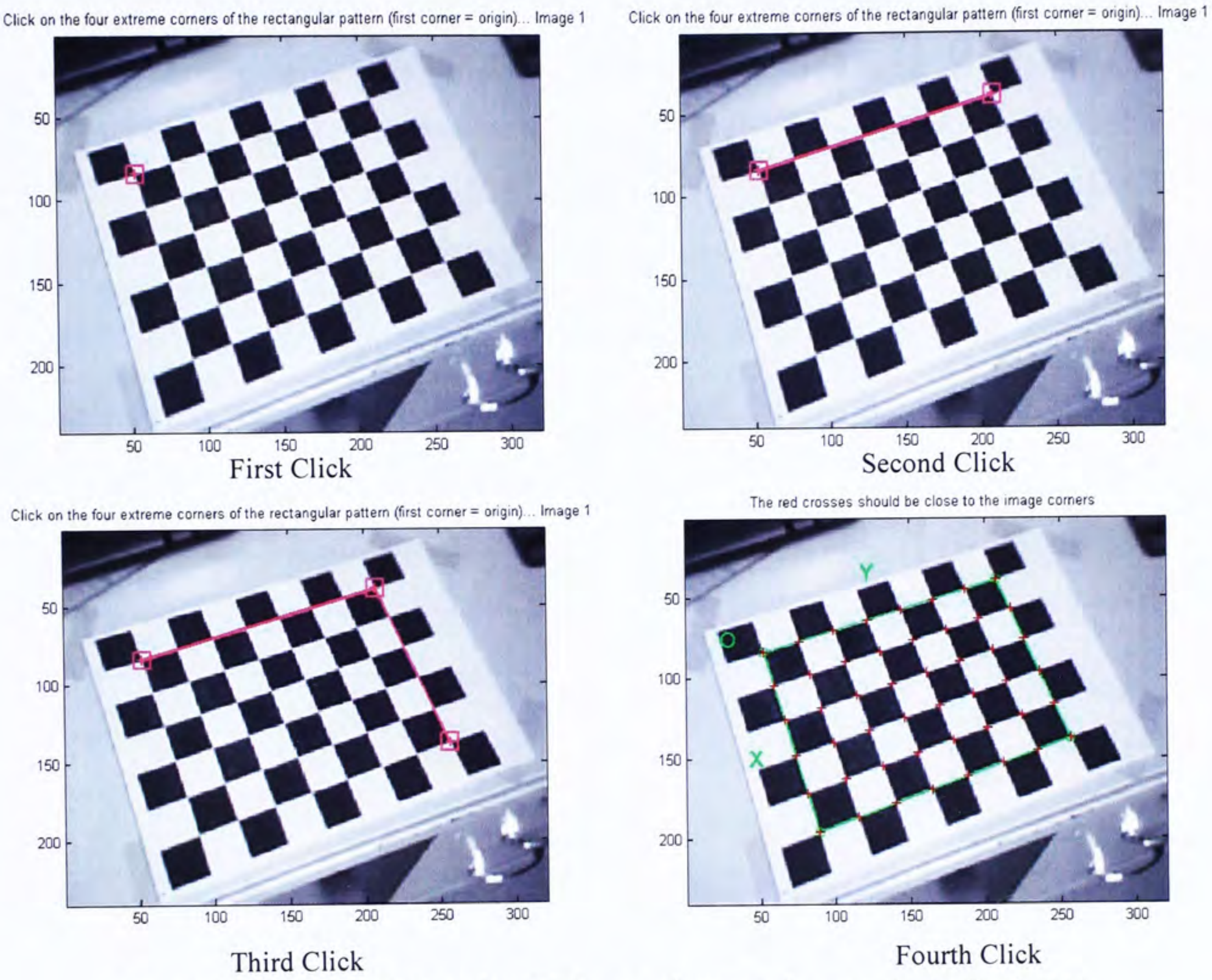


Figure 3.4 Images reading for camera calibration


Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1
50
100
150
200
50
100
150
200
250
300
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1
The red crosses should be close to the image corners
O
Y
X

First Click

Second Click

Third Click

Fourth Click

Figure 3.5 Definition of world frame and extracted corners in the first image

3.3.3 Main Calibration

After extracting the corners, this toolbox also provides us with a function to calibrate the camera. The algorithm takes the results from extracting corners as the input and gives both the intrinsic and extrinsic parameters as the output. Take the first image for example, Table 3.1 shows part of result of the 48 corners extraction which is the input for the calibration step.

Hence, the calibration can be performed and both intrinsic and extrinsic parameters are guaranteed [38], [39]. The basic process is, firstly, to print a pattern and attach it to a planar surface. Secondly, take several images of model plane under different orientations by moving the plane and the motion need not be known. Thirdly, detect the feature points in the images. The fourth step is to estimate the intrinsic and extrinsic parameters using homography theory and a closed-form solution. The fifth step is to estimate the coefficients of the radial distortion by solving the linear least squares. Finally, refine all parameters by a nonlinear minimizing Maximum Likelihood Estimation function [40]. After completing the iterations above, the results become convergent and dependable.

Table 3.1 Coordinates of the corners both in 3D world frame (mm) and 2D image frame (pixel)

X	Y	Z	u	v
0	210	0	206.5212	37.8179
30	210	0	215.5648	56.364
60	210	0	224.9023	75.4571
90	210	0	234.404	95.0896
120	210	0	244.8083	115.4566
150	210	0	255.6812	136.6592
0	180	0	185.7974	44.3233
30	180	0	194.3654	62.8391
60	180	0	203.4495	82.3228
⋮	⋮	⋮	⋮	⋮

Finally, the parameters are obtained and shown as follows:

$$KK = \begin{bmatrix} fc_1 & alpha\_c * fc_1 & cc_1 \\ 0 & fc_2 & cc_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 503.003 & 0 & 159.459 \\ 0 & 505.174 & 99.2532 \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(21)}$$

$$kc = \left[0.40694 \quad -1.93592 \quad -0.00608 \quad 0.00401 \quad 0.0000\right] \quad \textbf{(22)}$$

where KK is the camera matrix, the intrinsic parameters of the camera in linear model; $alpha\_c$ is the skew coefficient, which defines the angle between the x and y pixel axes, (in this experiment, it is equal to 0 which presents the orthogonal relationship between these two axes); focal length fc_1 and fc_2 are the same as ax and ay in (16); principal point cc_1 and cc_2 are the same as u_0 and v_0 in (10). The distortion coefficient kc has five components to describe the non-linear model. It contains both radial and decentering distortion coefficients by the first four components, which are equal to k_1, k_2, p_2 and p_1 respectively in (19) and (20), while the fifth is zero. All of these parameters referred to above are called the intrinsic parameters of the camera under the non-linear model.

As to the extrinsic parameters, there are some differences from the intrinsic ones. Since the intrinsic parameters present the structure of the camera itself, the results from the calibration will be consistent if no properties are adjusted. Different definitions of the world frame and the location of the camera mean that each image has its own relationship leading to inconsistent extrinsic parameters, so we will still take the first image for example. The rotation matrix and the translation vector are calculated as follows:

$$R_1 = \begin{bmatrix} 0.3582 & 0.9312 & 0.0673 \\ 0.8427 & -0.2914 & -0.4526 \\ -0.4019 & 0.2189 & -0.8892 \end{bmatrix} \qquad \textbf{(23)}$$

$$t_1 = [-133.1379 \quad -19.8033 \quad 629.6215]^T \qquad \textbf{(24)}$$

With the extrinsic parameters R_1 and t_1, the relationship between the camera frame and the world frame and the matrix M in the case of the first image can be calculated as below:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R_1 \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + t_1 \qquad \textbf{(25)}$$

$$M = \begin{bmatrix} 116.0891 & 503.302 & -107.9388 & 33430.0517 \\ 385.8203 & -125.4812 & -316.8977 & 52487.8364 \\ -0.4019 & 0.2189 & -0.8892 & 629.6215 \end{bmatrix} \qquad (26)$$

where M is implicit parameters which have no real physical meaning. After all the components are obtained, the basic work on camera calibration is completed. Similarly, the other nine matrices could also be established to describe their own properties and relationship.

3.4 Calibration Results of High Speed Camera

3.4.1 Lens Selection

As the high speed camera will need to operate at a high sampling rate to satisfy our demands, a suitable lens is also necessary to make sure the experiments are successful. The key-point for the lens is the size of its aperture. Among the parameters of the lens, the F parameter presents this property. The lower the F value, the bigger the aperture. In our test we made a comparison between the two lenses shown in Figure 3.6.

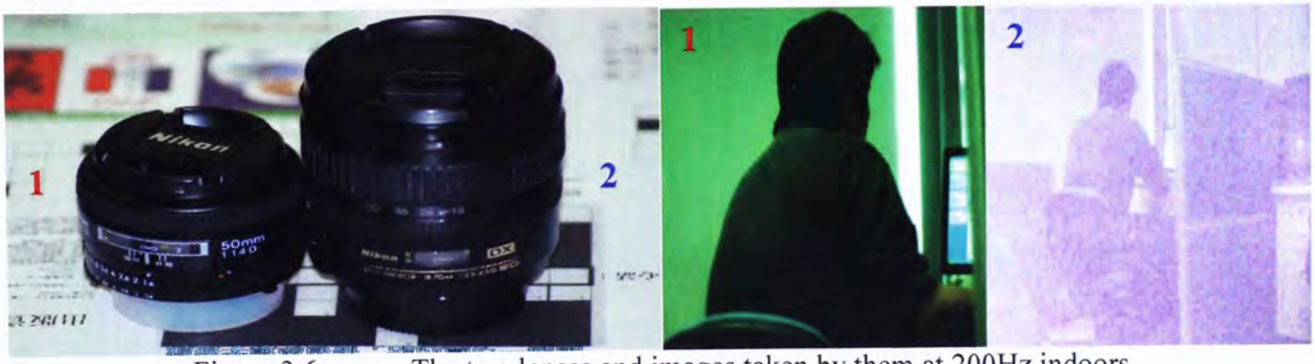
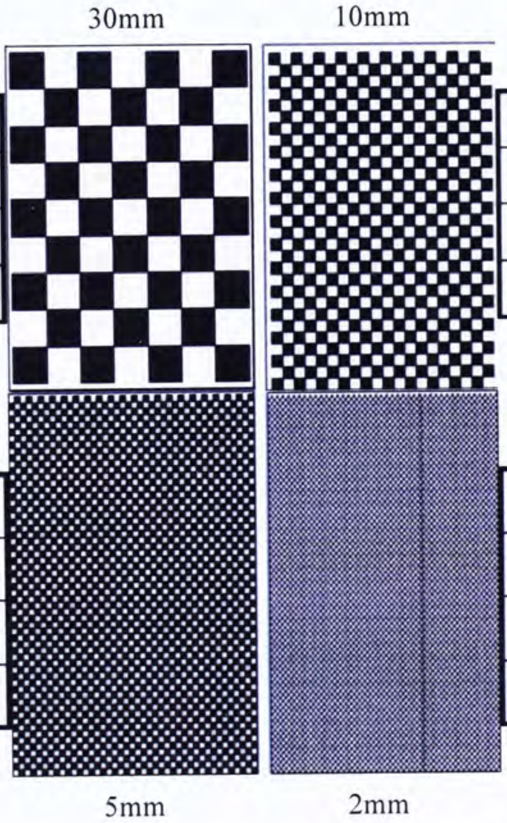


Figure 3.6 The two lenses and images taken by them at 200Hz indoors

Apparently, the left lens fits this situation much better. The main reason is that its F parameter is 1.4, which is much smaller than that of the lens on the right at 3.5-4.5. In other words, the left lens can absorb much more light than the right in the same environment. Also, the exposure time was set at 0.005s so that a lens with a bigger aperture can perform much better. As to the focus length, the left lens has a fixed focus of 50mm while the right one has an adjustable focus between 18 and 70mm, both of which would work in our experiments.

3.4.2 Property of High Speed Camera

In this test we are going to determine which size of grid on the template works best with the high speed camera (HSC). Since the previous template was made up of the grids of 30x30mm and the HSC has a higher resolution (maximum 1280×1024) than the digital camera we used previously, we are going to test other templates with grids of 10x10mm, 5x5mm and 2x2mm. All the templates and results are shown in Figure 3.7.



focus	4424	4434
error_f	1.74%	1.78%
Cpoint	763	444
Error_cp	7.35%	15.39%

focus	4424	4434
error_f	1.74%	1.78%
Cpoint	763	444
Error_cp	7.35%	15.39%

focus	4379	4395
error_f	0.25%	0.26%
Cpoint	770	440
Error_cp	0.96%	2.05%

focus	4323	4235
error_f	19.54%	19.72%
Cpoint	113	802
Error_cp	155.21%	24.7%

Figure 3.7 The calibration results of HSC with different templates

The calibration of the HSC with a template with a grid of 5x5mm can achieve the best result. As the more accurate parameters are very helpful for further calculation, like 3D reconstruction, this test helps us decide on a suitable template.

Another experiment to test the effect of the edge distortion of this camera had also been performed, which can tell us which area of the optical sensor has the least distortion. From Figure 3.8 we can conclude that HSC has a very good linear property and is suitable for further experiments. The reason for the differences between each focus may be related to the distance being inconsistent while the photos were being taken.

The four cases except the central one:

focus	4000+	4000+
error_f	less than 1%	less than 1%
Cpoint	690	327
Error_cp	0%	0%

The central case:

focus	4245	4250
error_f	0.55	0.55
Cpoint	659	275
Error_cp	0%	0%

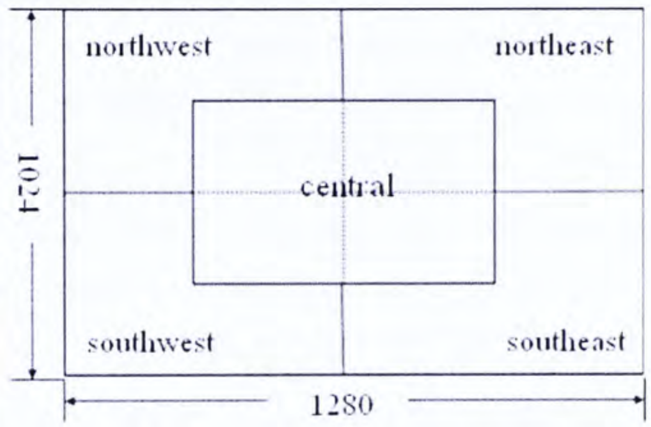


Figure 3.8 Calibration results of different areas in the optical sensor

Chapter 4

3D Attitude Calibration

4.1 The Necessity of Attitude Calibration

After matching, three-dimensional accelerations can be calculated but are only represented in the frame defined by us based on the matching plane. Since the output of our inertial sensors is described in the body reference frame, in order to calibrate the MAG-μIMU we need to perform a transformation between the accelerations we gained from matching and the ones from MAG-μIMU. This can be basically realized when an attitude calibration of the pen is completed. In fact, attitude calibration is based on the 3D position coordinates of two particular points on our digital pen. According to this information, the magnetometers of our MAG-μIMU are able to be calibrated. Additionally, it is feasible to estimate the three-dimensional angular velocities in the same way as we do for accelerations after we can get the information about the change of the attitude per frame. Hence, the entire nine-dimensional output of the MAG-μIMU can be rectified according to the more accurate data obtained from optical tracking [41].

Multiple Camera Calibration [42], [43] has been applied to 3D measurements for many years, so there exist many mature theories, successful experiments and dependable results. Derived from camera calibration, multiple camera calibration can solve many practical problems which single camera calibration cannot. In the next section, there is a very brief introduction to the concept of this technique.

4.2 Stereo Vision and 3D Reconstruction

Stereo vision is a method to capture three-dimensional geometric information through multiple images (generally two). As has been known for a long time, almost all the creatures with vision ability have two eyes. That is because observation with two eyes can obtain the information of depth which makes the sense of sight work much better. To carry out 3D reconstruction by the stereo vision method in the computer vision field is to calculate out the 3D geometric information of the object based on two or more 2D images of it. In Figure 4.1, if there is only one camera to observe the space point P, we cannot gain the exact information to describe its position as introduced in the previous chapter. The situation can be changed if there is another camera to cooperate.

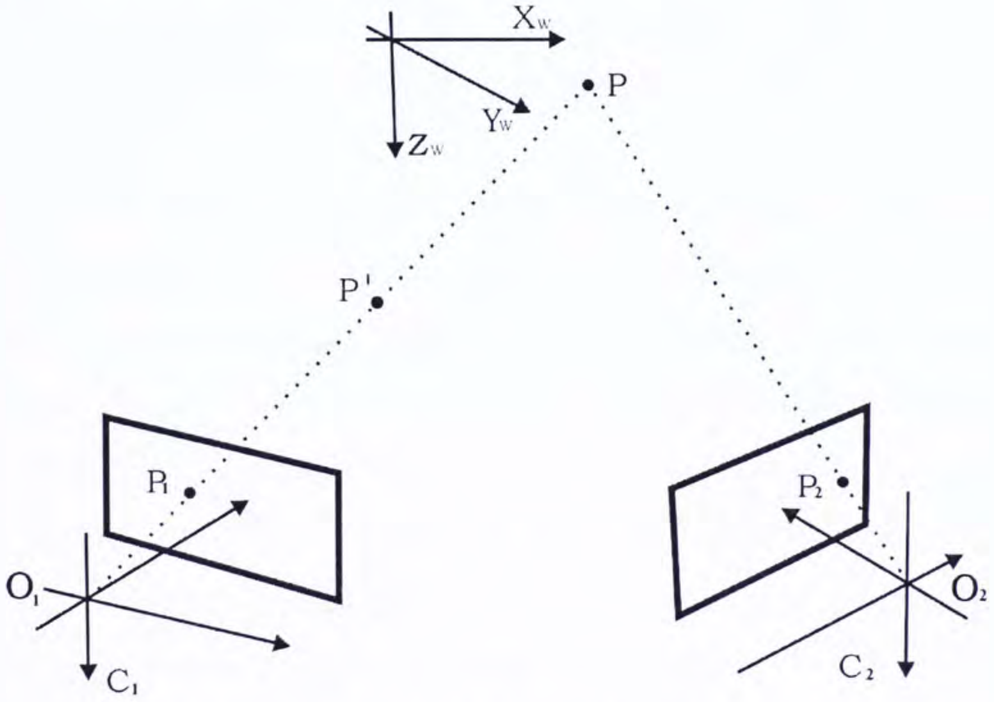


Figure 4.1 Observation of a space point with two cameras

4.2.1 Physical Meaning and Mathematical Model Proof

When we use two cameras to observe this point p at the same time and make sure both the image points p_1 and p_2 belong to P, it is possible to specify the coordinates of it. As known, point P lies on not only O_1p_1 but also O_2p_2. Hence, point P is supposed to be the intersection of the lines O_1p_1 and O_2p_2 and there is only one solution of its 3D coordinates. This is the basic theory of stereo vision.

As to the calculation, what conditions should be satisfied to solve the 3D coordinates? Assuming that the relative points p_1 and p_2 of P in two images are known and the two cameras are calibrated individually already, from equation (16) four linear equations about X_w, Y_w and Z_w can be obtained after cancelling Z_{c1} and Z_{c2} as follows:

$$\begin{aligned}(u_1m_{31}^1 - m_{11}^1)X_w + (u_1m_{32}^1 - m_{12}^1)Y_w + (u_1m_{33}^1 - m_{13}^1)Z_w &= m_{14}^1 - u_1m_{34}^1 \\ (v_1m_{31}^1 - m_{21}^1)X_w + (v_1m_{32}^1 - m_{22}^1)Y_w + (v_1m_{33}^1 - m_{23}^1)Z_w &= m_{24}^1 - v_1m_{34}^1\end{aligned} \qquad (27)$$

$$\begin{aligned}(u_2m_{31}^2 - m_{11}^2)X_w + (u_2m_{32}^2 - m_{12}^2)Y_w + (u_2m_{33}^2 - m_{13}^2)Z_w &= m_{14}^2 - u_2m_{34}^2 \\ (v_2m_{31}^2 - m_{21}^2)X_w + (v_2m_{32}^2 - m_{22}^2)Y_w + (v_2m_{33}^2 - m_{23}^2)Z_w &= m_{24}^2 - v_2m_{34}^2\end{aligned} \qquad (28)$$

where m^1 and m^2 are the components of the M matrix belonging to the two cameras respectively; u, v are the image coordinates of the relative points p_1, p_2. According to analytic geometry, the function of a plane in 3D space is linear and a substitution of two plane functions can result in a line function, which presents the intersection line of the two planes. Equations (27) and (28) have the geometric meaning to present the lines O_1p_1 and O_2p_2. As P is the intersection of them, the coordinates (X_w, Y_w, Z_w) must satisfy both (27) and (28). Finally, we can get the solution of the coordinates easily because there are four linear equations about three unknown variables.

4.2.2 3D Point Reconstruction

Based on single camera calibration, multiple camera calibration as shown in Figure 4.2 can provide us with all the helpful parameters of both cameras. In addition, the same template should be used during the experiment, which means that the same plane, the same time and the same world frame are defined for the two cameras. According to the same world frame and the extrinsic parameters of each camera, the relative position between the right and left camera frame is obtained. It can be presented as follows:

$$X_{CR} = R' \cdot X_{CL} + T \tag{29}$$

where X_{CR} and X_{CL} are the 3D coordinates vectors in each camera frame; R' is the rotation matrix and T is the translation vector, which present the relationship between the two camera frames. Then we can freely transform any point described in the right camera frame into the left one, or from the left to the right, according to this equation.

In the Camera Calibration Toolbox, the above information can be obtained by a function named 'stereo calibration' after all parameters of both cameras are loaded. Also it provides us with another function named '*stereo triangulation*', which can make use of the result about R' and T to get the relationship between the coordinates of the image frames and the camera frames:

$$\begin{aligned}&[Xc\_1\_left, Xc\_1\_right] = stereo\_triangulation \\ &(x\_left\_1, x\_right\_1, om, T, M_{1L}, M_{1R})\end{aligned} \tag{30}$$

where x_left_1 and x_right_1 are 2D pixel coordinates of the points which are going to be reconstructed in the left and right images respectively; om is the character vector of the rotation matrix R', which satisfies the relationship $R' = rodrigues\,(om)$; T is the translation vector; M_{1L} and M_{1R} are the intrinsic parameters matrix of the cameras; and the output Xc_1_left and Xc_1_right are the 3D coordinates described in the left and right camera frames.

This means that any 2D image point we pick up from pictures can be matched to a 3D point described in the camera frame. In other words, when any image coordinates vector is input, a corresponding 3D point in the camera frame can be guaranteed as the output. Then a point in the world frame can be calculated as follows:

$$Xw = R^{-1} \cdot (Xc - t) \tag{31}$$

where Xw and Xc are the 3D coordinates vectors (X_w, Y_w, Z_w) and (X_c, Y_c, Z_c) in the world frame and the camera frame, respectively. In this case, it is feasible to estimate the position of the UDWI during writing. Besides that, the attitude can be calculated from reconstructing two points on the UDWI, which are kept consistent during the experiment.

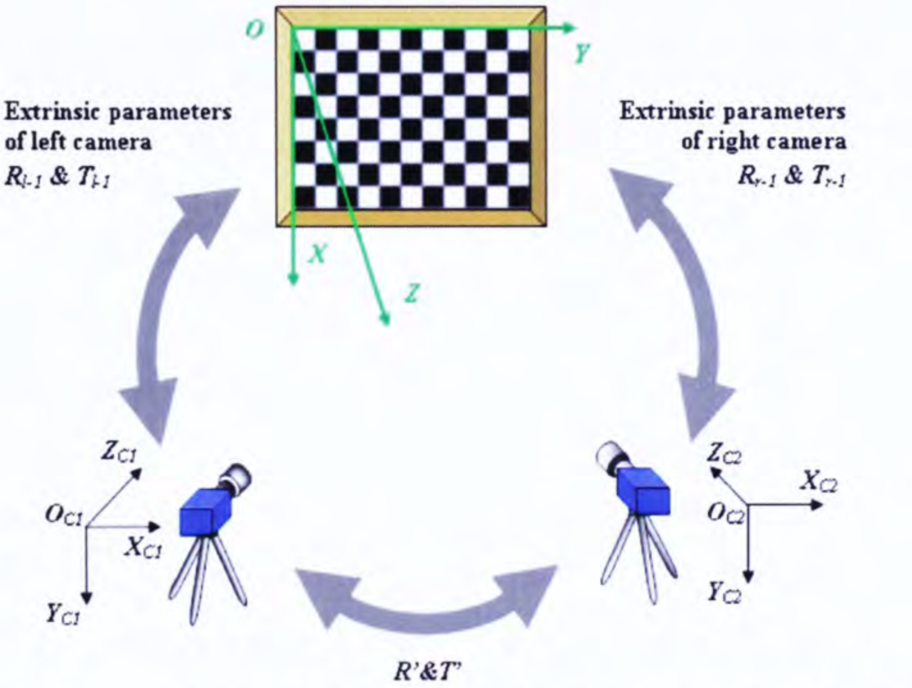


Figure 4.2 Schematics of multiple camera calibration experimental setup

4.3 Example of 3D Point Reconstruction

With the method illustrated above, we did a test to reconstruct a 3D point. Figure 4.3 shows all the images used to carry out the multiple camera calibration taken by two cameras. We describe the different cameras as the left camera and the right camera. There are five pairs of images taken by the left camera and the right camera respectively, and each pair was captured at the same time. But the distance and the orientation were different between the template and the cameras each pair while the relative position between the two cameras was fixed throughout the experiment. Firstly, we calibrated the cameras individually through the images and both the intrinsic and extrinsic parameters were ascertained.

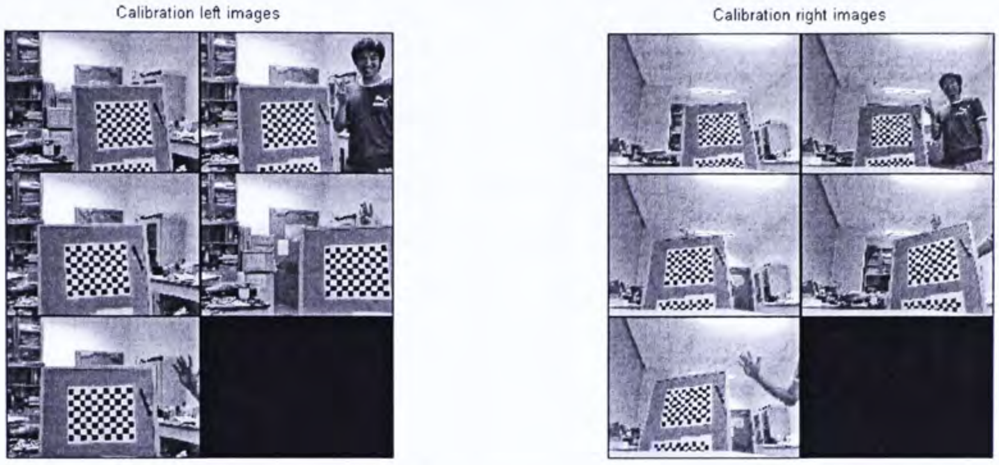


Figure 4.3 Images for stereo calibration

Secondly, we utilized the function *stereo_gui* on the toolbox, then loaded the files containing the parameters of the two cameras into this stereo camera calibration system. Then the necessary parameters R', om and T could be calculated after running the stereo calibration. Thirdly, we figured out a test point in the second image taken by the left camera. Actually, any point in every image taken by the both cameras could be reconstructed. In this test we selected a particular one which was not only one of the corners but also the origin with the coordinates $(0 \quad 0 \quad 0)$ in the world frame. The coordinates of the corners, both the 3D in the world frame and the 2D in the image frame, could be found easily after calibration. Figure 4.4 illustrates the point which was going to be reconstructed.

Next, making use of the formula shown in equation (30), the coordinates in both cameras' frames were ascertained. Finally, with correct parameters R and t, the coordinates in the world frame of this image point were obtained in the same way as shown in equation (31). Similarly, another neighboring corner of the origin was also reconstructed, of which the coordinates in the world frame were supposed to be $(30 \quad 0 \quad 0)$. All the parameters and the final results are presented in Table 4.1.

Among all the components in Table 4.1, only *Xw* has its own meaningful unit in millimeters (mm). The error range is with $\left[-2 \quad 2\right]$ in mm. In fact, there were very small differences of the image plane information between the defined values and the measured ones. As to the depth information, the error was obviously bigger. All the errors were introduced probably because of the errors in camera calibration and the distortion of the lenses. In this test, the left camera was a digital camera with a relatively higher quality than the right one, a web camera. Consequently, the results from the left system were more reliable and accurate.

4.4 Idea of Attitude Calibration

Reconstruction of a 3D point in the particular frame has been realized. Hence, if we select two special points on the surface of the UDWI and reconstruct them, we can get a line which can present the location and attitude information (see Figure 4.5). The angles between the UWDI and the three axes of the frames can also be calculated. Assuming that based on the motion estimation with video compression aid technique, a similar algorithm to the matching one can be developed to capture the image information of the two special points, it has the potential to obtain the location and attitude information through multiple camera calibration, which can be synchronized to calibrate the IMU.

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} angle\_x_{n+1} - angle\_x_n \\ angle\_y_{n+1} - angle\_y_n \\ angle\_z_{n+1} - angle\_z_n \end{bmatrix} \qquad (32)$$

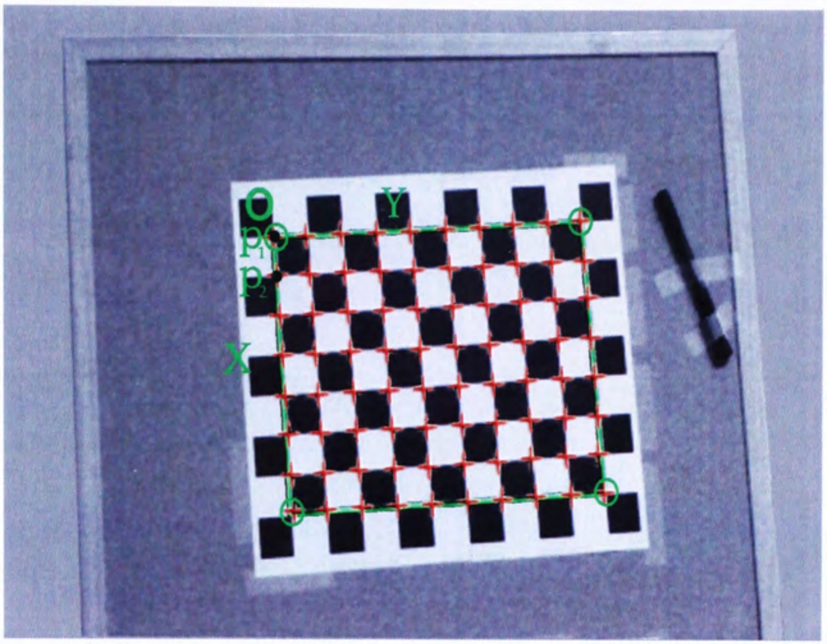


Figure 4.4 Points to be reconstructed in the image

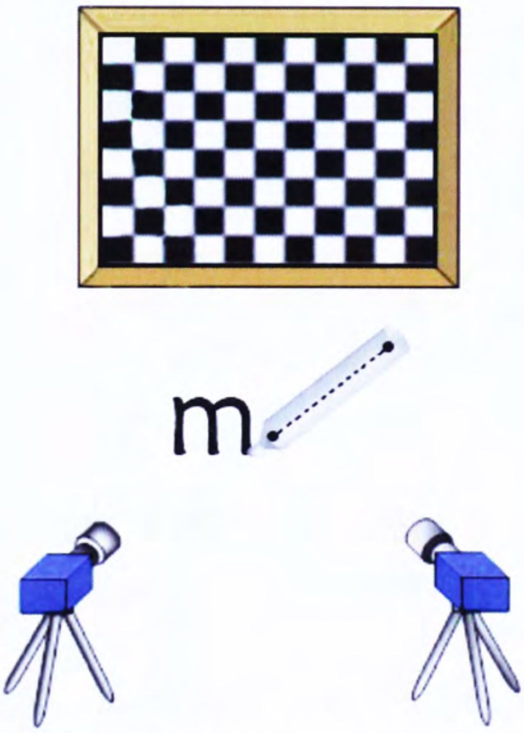


Figure 4.5 Schematic of attitude calibration

Table 4.1 Experimental parameters and results of 3D point reconstruction

Parameter	Point 1	Point 2
(u,v) in left	$(216.43,263.9343)$	$(217.5147,280.0521)$
(u,v) in right	$(246.7543,296.6581)$	$(246.6414,307.408)$
om	$[-0.3579 \quad 0.2708 \quad -0.0669]$	
R'	$\begin{bmatrix} 0.9618 & 0.017 & 0.2734 \\ -0.1123 & 0.9348 & 0.3368 \\ -0.2498 & -0.3546 & 0.901 \end{bmatrix}$	
T	$[-529.9771 \quad -503.0145 \quad -274.8245]$	
KK in left	$\begin{bmatrix} 1037.2632 & 0 & 323.447 \\ 0 & 1035.5217 & 245.1711 \\ 0 & 0 & 1 \end{bmatrix}$	
kc in left	$[0.039 \quad -0.0872 \quad 0.0047 \quad 0.0041 \quad 0]$	
KK in right	$\begin{bmatrix} 552.6138 & 0 & 319.8192 \\ 0 & 546.4409 & 224.8596 \\ 0 & 0 & 1 \end{bmatrix}$	
kc in right	$[0.0624 \quad -0.2748 \quad -0.0047 \quad 0.0015 \quad 0]$	
R in left	$\begin{bmatrix} 0.0853 & 0.9628 & 0.2563 \\ 0.9807 & -0.0357 & -0.192 \\ -0.1757 & 0.2677 & -0.9473 \end{bmatrix}$	
t in left	$[-197.6369 \quad 34.7595 \quad 1913.2074]$	
Results	Ponit1	Point2
Xc in left	$(-197.3321 \quad 34.4745 \quad 1912.1277)^T$	$(-195.2877 \quad 64.4289 \quad 1909.4584)^T$
Xw in left	$(-0.0637 \quad 0.0146 \quad 1.1557)^T$	$(29.9571 \quad 0.1976 \quad -1.5437)^T$

Chapter 5

Experimental Results

5.1 Calculation of Proportional Parameter

An important function of the marker on the transparency is to calculate the proportional parameter K, which is necessary for the transformation between pixel and physical units. From Figure 2.5, it is very clear that the points on the marker have very different gray-scale value from those of nearby points. Thus we can make use of this property to develop a method. Figure 5.1 shows the process of determining the parameter before we deal with the data from matching.

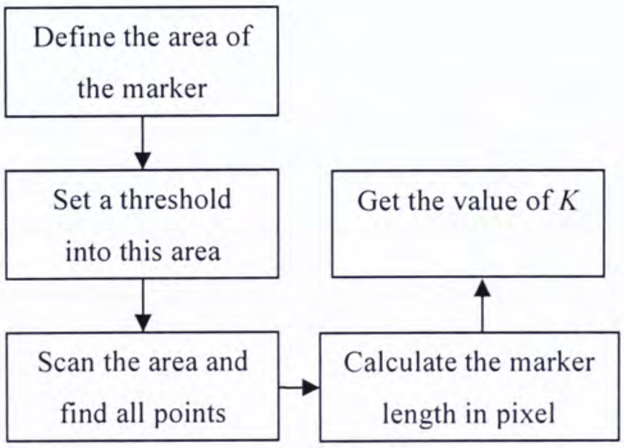
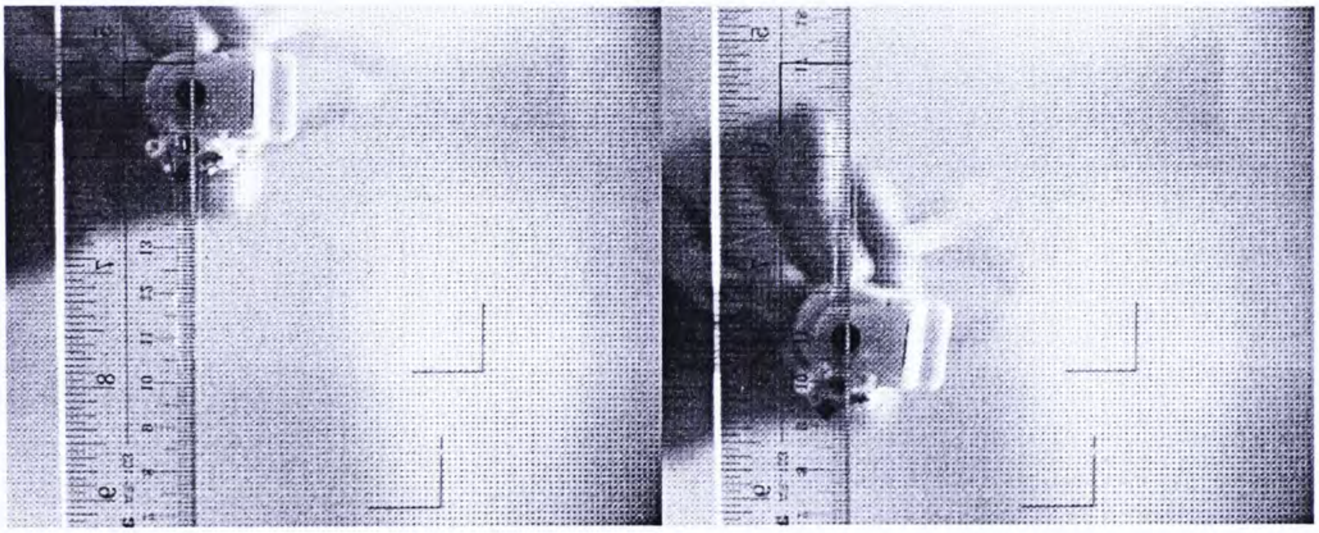

Define the area of the marker
Set a threshold into this area
Scan the area and find all points
Calculate the marker length in pixel
Get the value of *K*

Figure 5.1 Flow chart on the calculation of proportional parameter

5.2 Accuracy Test of Stroke Reconstruction

In order to ensure the accuracy and reliability of the tracking reconstruction based on our matching system, a test has been completed. In this test we drew a straight line with a ruler aid as shown in Figure 5.2. In this case we could read how long the line was as the reference and then another conclusion about this length could be obtained from the vision system. Finally, a comparison was made and a satisfactory result was obtained.



(a) (b)

Figure 5.2 Image sequence of straight line for test (a) the first image (b) the last image

From the images above we could read that the original position was at 16.3cm when we used the central point of the pen-tip circle to present the position of the pen on the image plane. The end point was at 11.0cm. Hence the line was supposed to be 53mm long. In addition, based on the method introduced in Section **5.1** the proportional parameter *K* was equal to 1.163575e^{-4} meter per pixel. Then we used our matching algorithm with this image sequence to reconstruct this line and examined this.

Figure 5.3 shows the reconstructed line through the matching algorithm. The unit was all in meters transformed by the parameter *K*. Besides that, we had set the beginning point as the origin so the initial point was at $(0 \quad 0)$ and the end point was at $(0 \quad -0.05294)$. Consequently, the length of this reconstructed line was 52.94mm. Compared with the other length measured by ruler, the difference was 0.06mm and error was about 0.11%. With such good performance, we think this vision calibration system is qualified to calibrate the 2D motion.

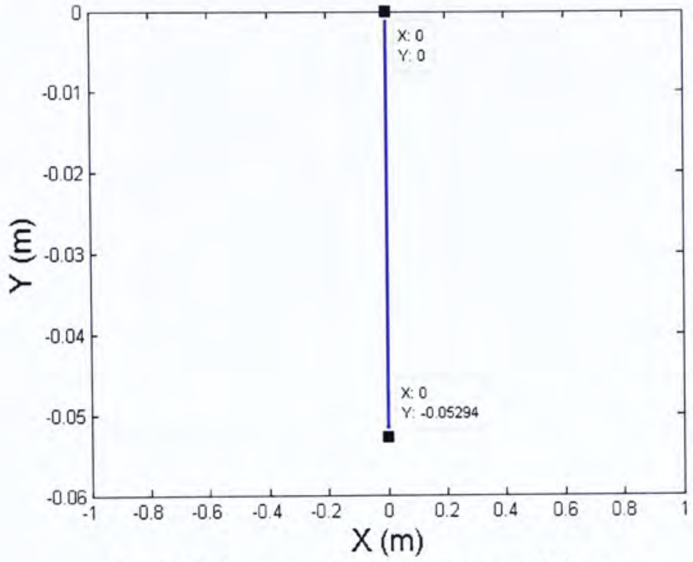


Figure 5.3 Line reconstructed for test

5.3 Writing Experiments of 26 Letters

In order to analyze the properties of our MEMS inertial sensors, the writing experiments were carried out. All 26 letters of the English alphabet were written in lower-case with the pen illustrated in Figure 5.4. During the motion of writing each letter, the attitude of the pen was ideally kept consistent and every letter was finished in a single stroke. Hence, as introduced in the discussion above, two independent groups of motion data were able to be obtained from the sensor output and the optical tracking system respectively. Consequently, a comparison between them could be achieved which will be helpful for us to overcome the drift better in the future. In view of the limits of word-length for this dissertation, three sets of experimental results will be introduced in this section, while the reminder can be found in the Appendix. All the curves for velocity and acceleration in the comparisons below consist of points, whose number depend on the time recorded to write a letter, and the interval between two neighboring points is 0.005s.

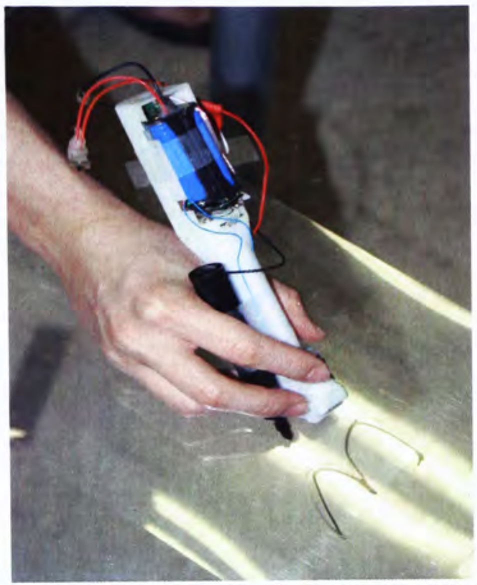


Figure 5.4 The writing instrument with a marker

5.3.1 Experimental Results of Letter *b*

In this subsection, all the data obtained from writing the letter *b* in a single stroke will be presented in the form of comparison, including 2D accelerations, 2D velocities and final characters. First of all, let us compare the resulting images. Figure 5.5 shows the last image of the sequence, which represents exactly the appearance of the hand-written letter while the letter found through the matching algorithm is presented in Figure 5.6. This found letter is made up of 165 blocks, which can tell us that the writing motion lasted approximately 0.82s.

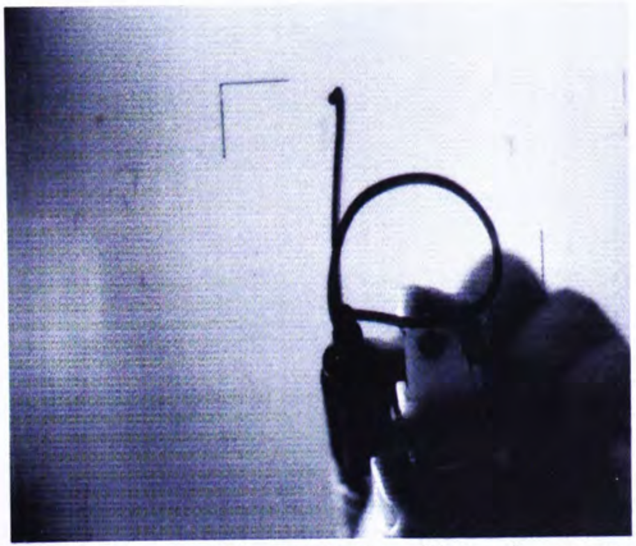


Figure 5.5 The last image of letter *b* sequence

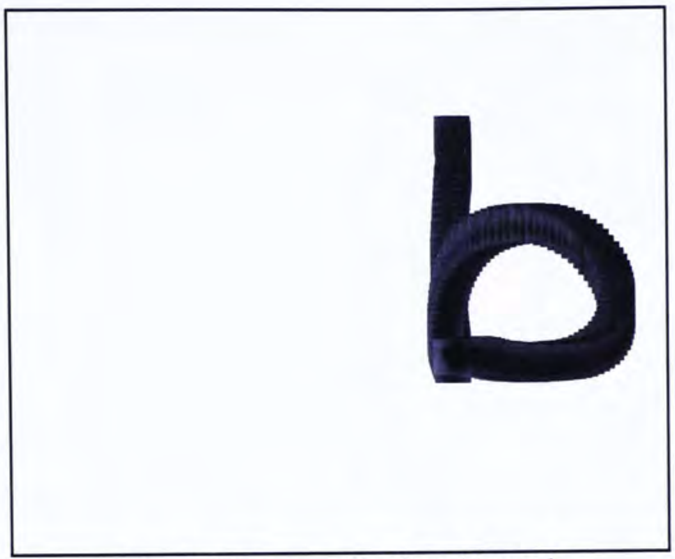


Figure 5.6 The found letter *b*

Based on the observation that the found character is almost the same as the written one, we could conduct the calculation of the accelerations and velocities from the position information after matching. On the other hand, the sensor output received through the Bluetooth Module was also processed and transformed the corresponding information. Figure 5.7 and Figure 5.8 illustrate the comparisons of the accelerations and the velocities in both *X* and *Y* axes, respectively. In this chapter, all the curves in green stand for the sensor output while the red ones are from the optical and all the variables are uniformed in physical units.

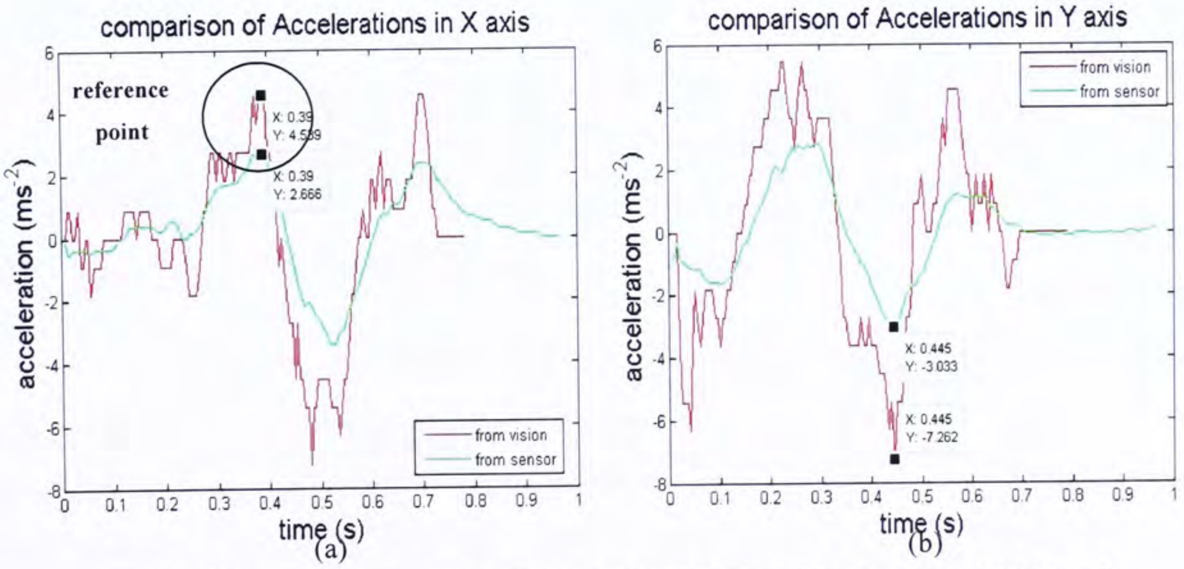
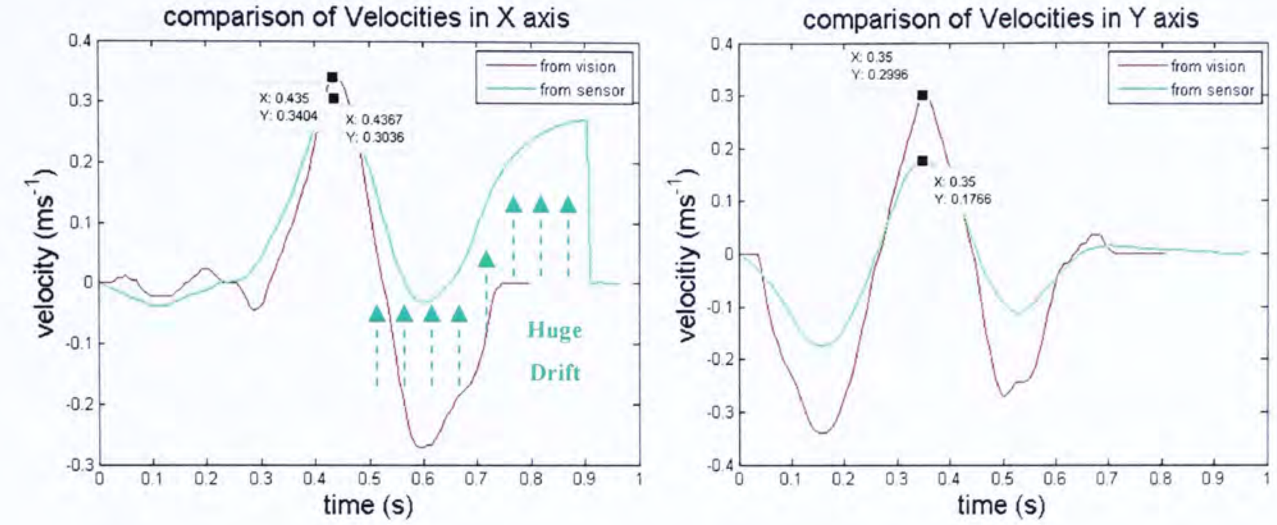


Figure 5.7 Comparisons of accelerations of letter *b* (a) in *X* axis (b) in *Y* axis



(a) (b)

Figure 5.8 Comparisons of velocities of letter *b* (a) in *X* axis (b) in *Y* axis

After the accelerometer signal was transformed, the accelerations were calculated out first. From the figures above, the acceleration curves obtained in two ways matched well with each other. However, it became obvious there was a difference in the comparisons of the velocities, especially the ones along the *X* axis (see Figure 5.8 (a)). In the area specified by a dashed line, there was a significant difference between the data curve and the vision one. The green curve drifted towards the positive orientation of the *X* axis at its second half while the red curve was considered as a reference. This phenomenon was caused by the intrinsic drift and noise of the accelerometers when their output underwent integration from acceleration to velocity. Hence, this effect definitely prevented us from drawing this character exactly with the sensor output. Figure 5.9 illustrates the comparison of the letters. It is clear that the corresponding drift existed and might even become worse because the signal of the accelerometers eventually underwent a second order of integration from acceleration to displacement.

Since the sample rate is very high in this experiment, we consider the peak, which has the most obvious motion characteristics, as the reference point to synchronize the two curves. The derivation orientations of the two methods are inverse to each other. According to the reference data from the optical tracking system, it is possible to estimate how the sensor drift works, which can probably cause it difficulty in drawing a character through the sensor output directly. On the other hand, data provided by the optical tracking system performed much more accurately to reconstruct a letter and could be considered as a good reference.

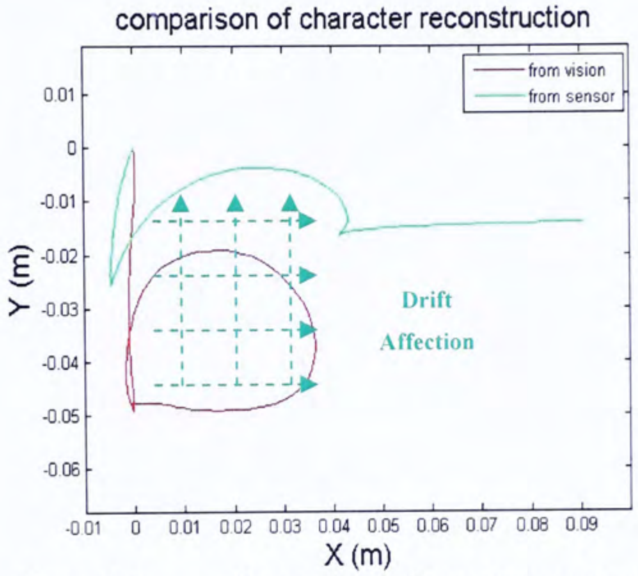


Figure 5.9 Comparison of letter *b* reconstructed

5.3.2 Experimental Results of Letter *n* with ZVC

Let us take another letter for example, letter *n*. Figure 5.10 and Figure 5.11 illustrate the written and the found letters and Figure 5.12 shows the comparisons of accelerations in both *X* and *Y* axes.

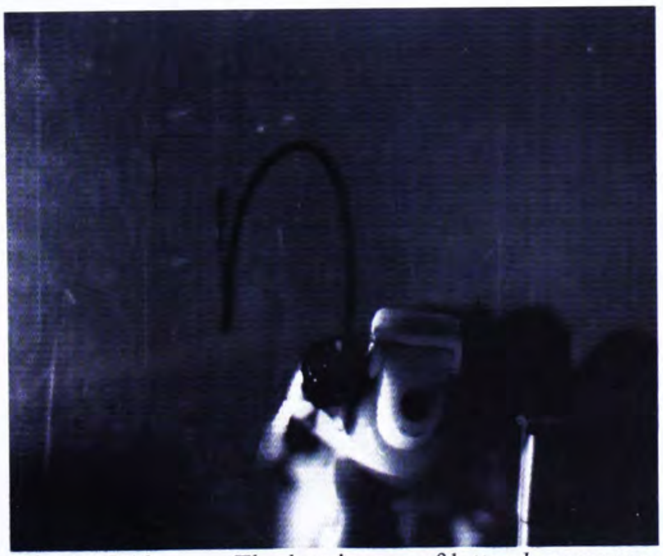


Figure 5.10 The last image of letter *b* sequence

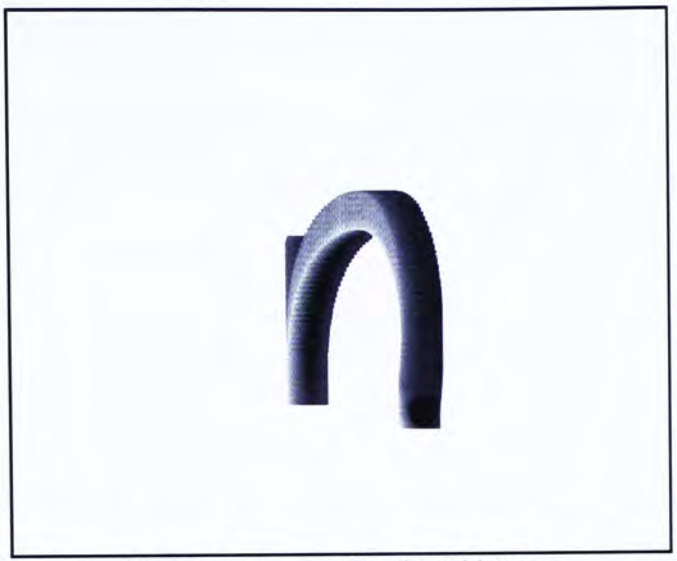
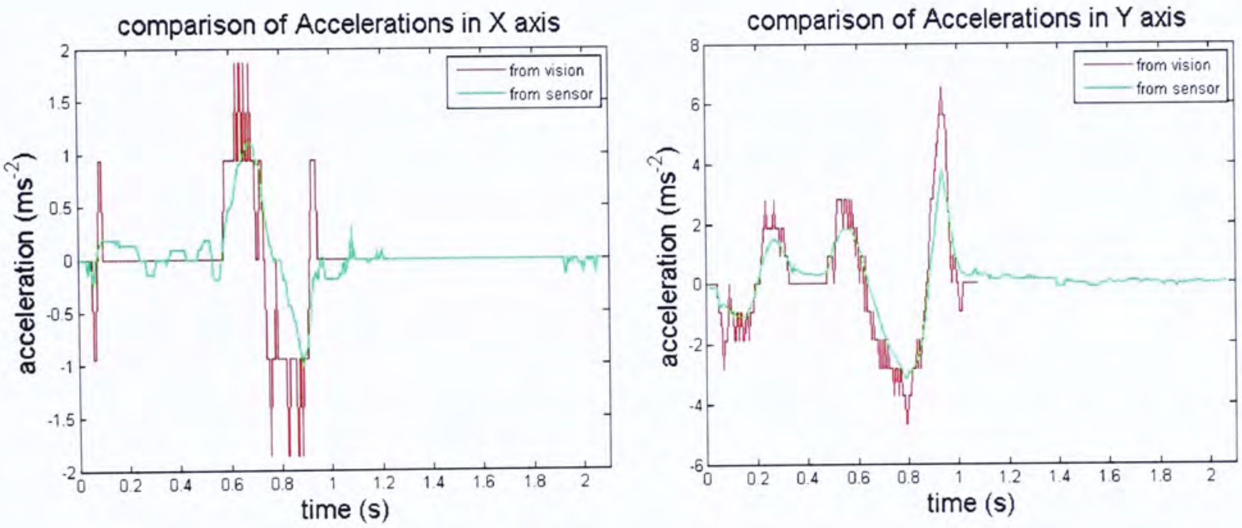


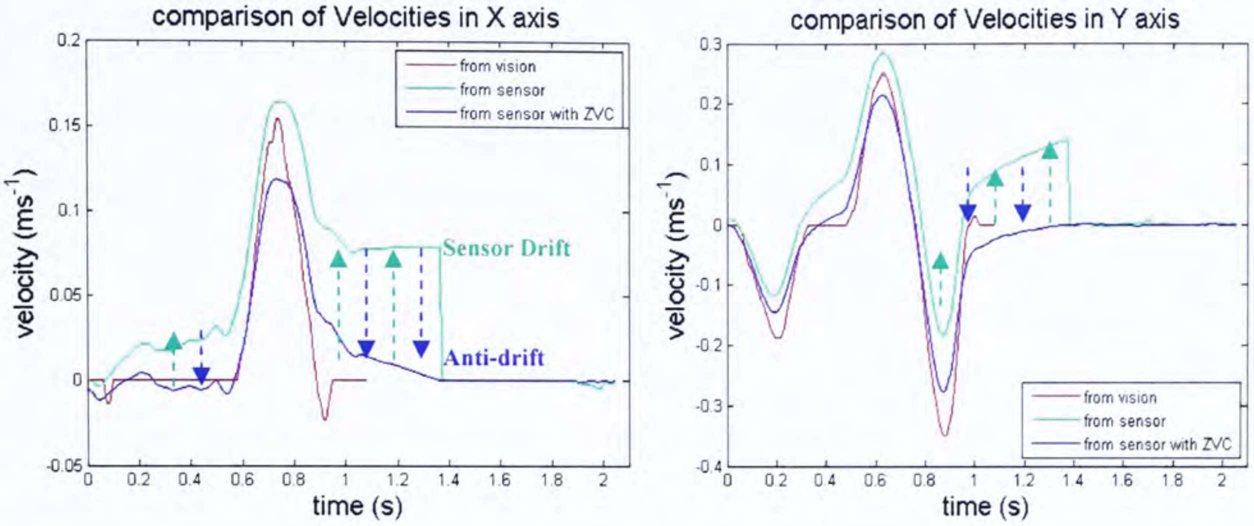
Figure 5.11 The found letter *n*



(a) (b)

Figure 5.12 Comparisons of accelerations of letter *n* (a) in *X* axis (b) in *Y* axis

Owing to the sensor's intrinsic drift, which occurred when the accelerometer's output underwent an integration and prevented us from reconstructing the letter, an algorithm of compensation, Zero Velocity Compensation (ZVC) [44], was added into the algorithm based on the sensor output. This compensation algorithm, together with a stroke segmentation algorithm [45], is supposed to overcome the drift better and to be useful to gain more accurate velocity information. After our algorithm for sensors was improved based on this technique, new curves were obtained, as shown in blue in Figure 5.13 and Figure 5.14.



(a) (b)

Figure 5.13 Comparisons of velocities of letter *n* (a) in *X* axis (b) in *Y* axis

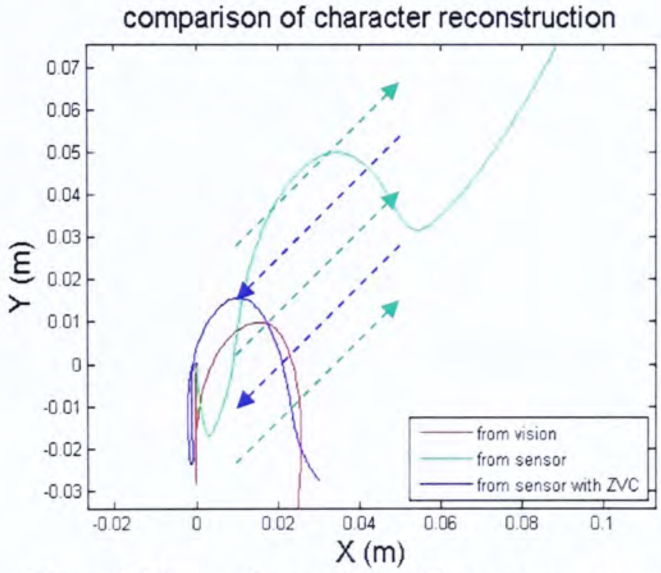
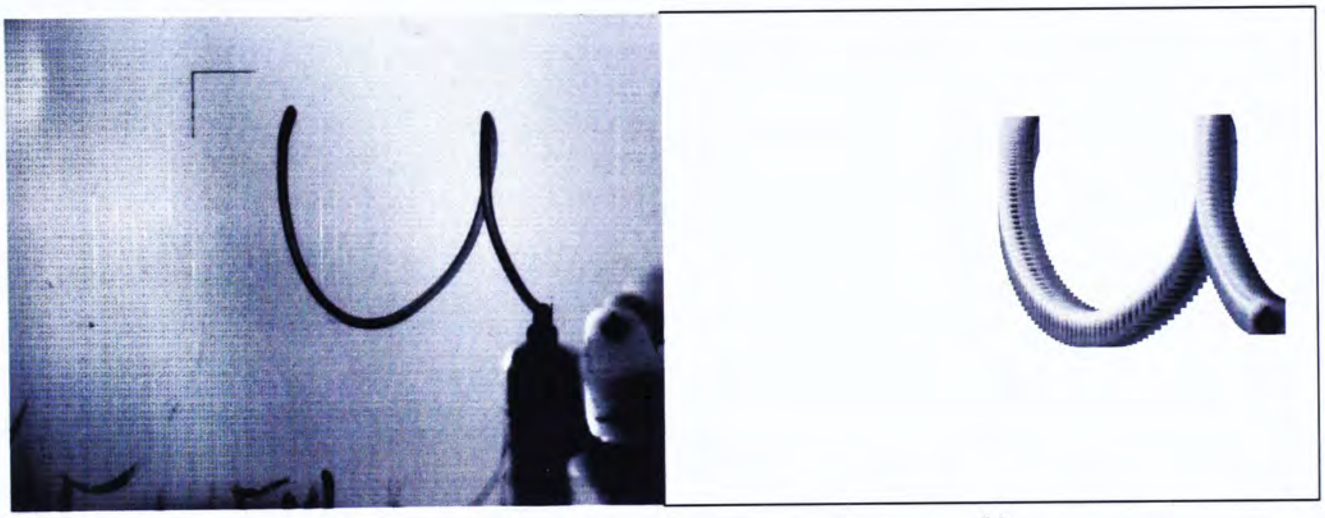


Figure 5.14 Comparison of letter *n* reconstructed

From Figure 5.13, the sensor drift still obviously existed. However, the curves obtained with the aid of ZVC (in blue) overcame this drawback significantly in both axes. Hence, a much better result could be obtained from the velocity information more similar to that from the optical system. Even though the ZVC algorithm cannot work successfully all the time in the current circumstances, the phenomenon that the more similar the curves from sensor output are to those from the optical system, the better characters we get, further proves that the optical tracking system is qualified to provide us with a more accurate reference.

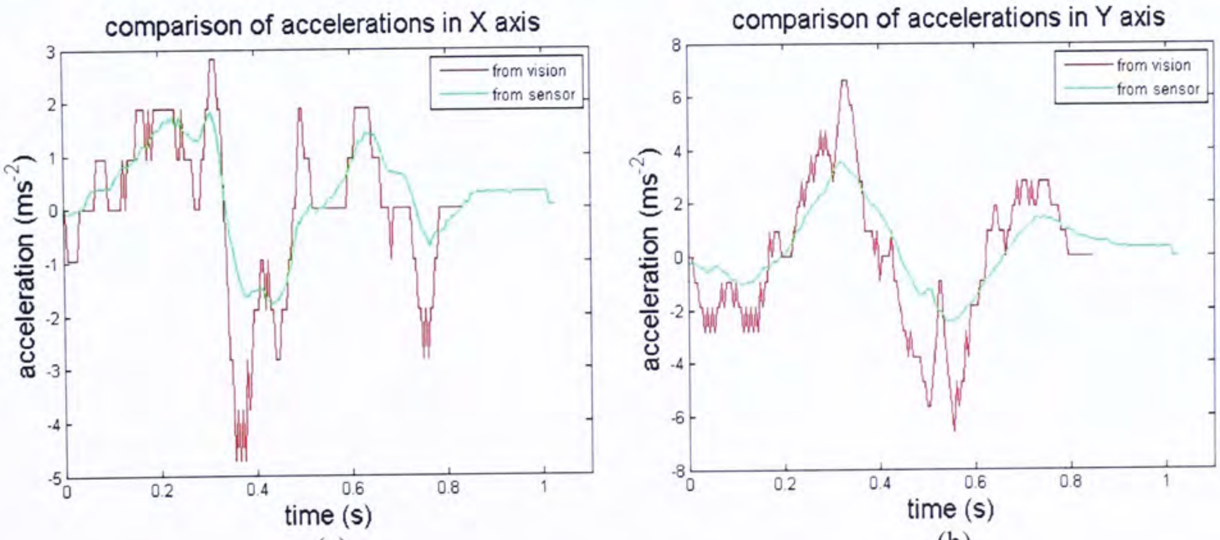
5.3.3 Experimental Results of Letter *u*

Another set of experiments will be described here, for the letter *u*. The time taken to write this *u* was about 0.85s and the reconstructed one was made up of 170 blocks. Figure 5.15 illustrates both the written letter and the reconstructed one after matching. The comparisons of accelerations in both the *x* and *y* axis are shown in Figure 5.16; while the comparisons of velocities, including ZVC algorithm, are contained in Figure 5.17. Finally, Figure 5.18 presents the figurations from optical tracking, sensor output and sensor output with the aid of the ZVC algorithm, respectively.



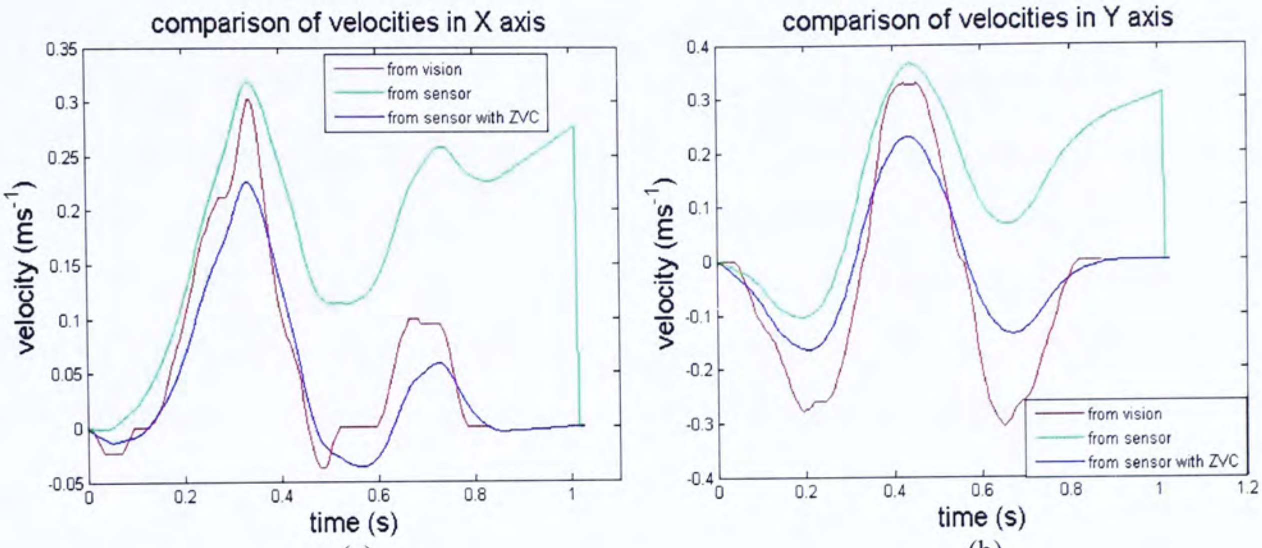
(a) (b)

Figure 5.15 The letter *u* (a) the last image of *u* sequence (b) the found *u*



(a) (b)

Figure 5.16 Comparisons of accelerations of letter *n* (a) in *X* axis (b) in *Y* axis



(a)

(b)

Figure 5.17 Comparisons of velocities of letter *n* (a) in *X* axis (b) in *Y* axis

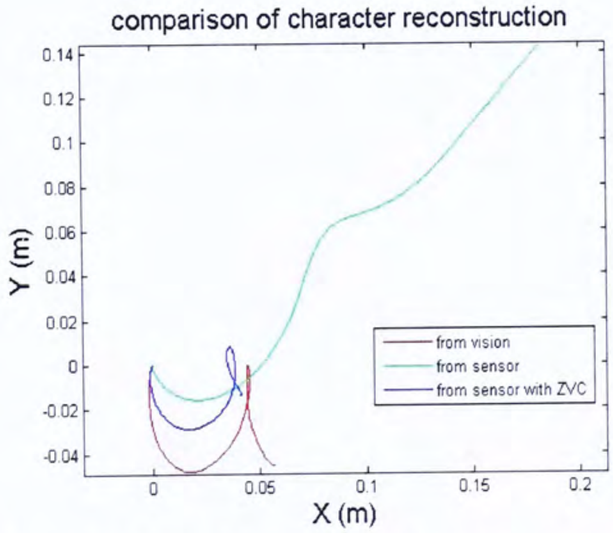


Figure 5.18 Comparison of letter *u* reconstructed

5.4 Writing of Single Letter *s* – Multiple Tests

In order to test the stability of the sensors, the letter *s* was written eight times. The attitude of the pen was kept as consistent as possible while it was moving to write. Figure 5.19 illustrates all eight reconstructed results through the optical tracking system. For the first two instances, since they were written before the other six, there were some differences between them in shape. As to the remaining six examples, we tried to write the letter in the same track, which would make it more convenient for us to compare the sensor output.

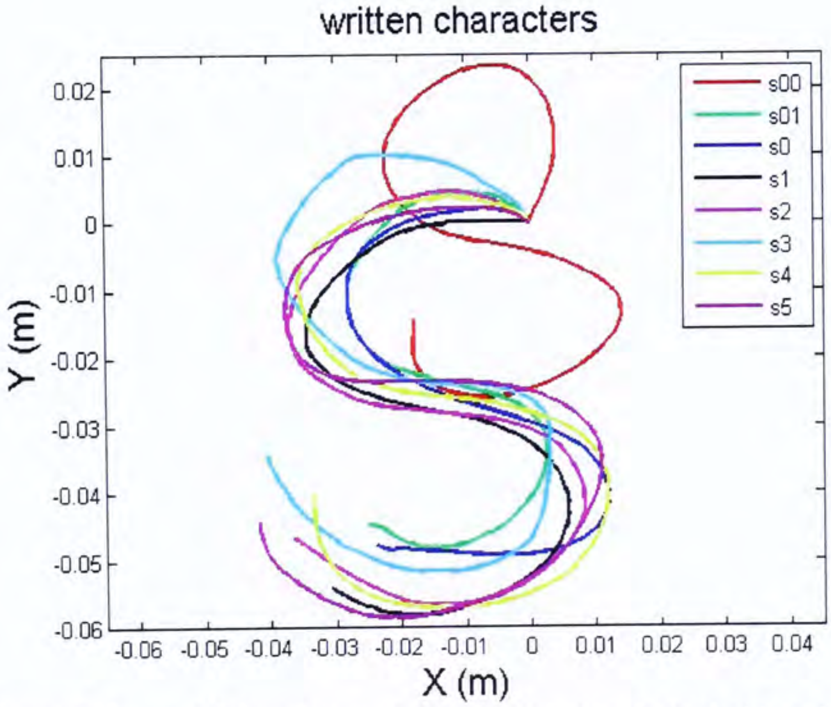


Figure 5.19 Written letter *s* from optical system and matching algorithm

Let us now examine the results from the sensor output to determine whether the sensor output gave similar information when the pen was writing the same letter. Figure 5.20 shows the final results of the sensor output and Figure 5.21 presents those with ZVC.

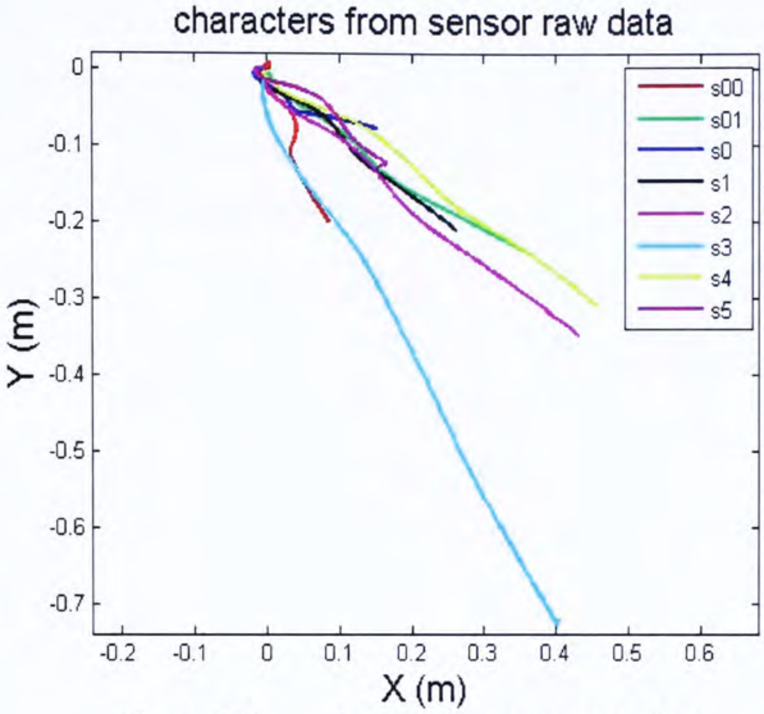


Figure 5.20 Letter *s* from sensor raw data

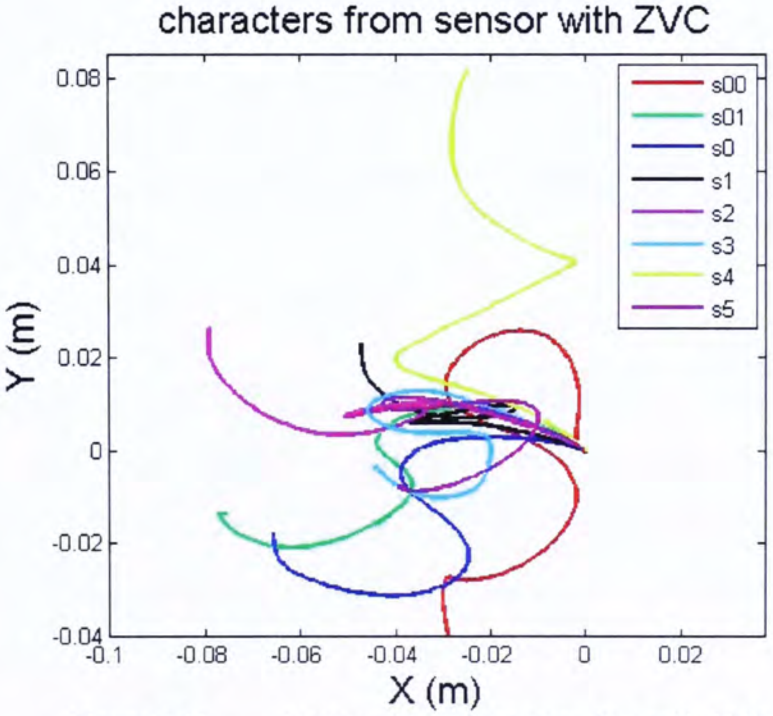


Figure 5.21 Letter *s* from sensor raw data with ZVC

From the figures above, we can conclude that the sensor output was not stable enough to reconstruct the same moving track after several similar motions were completed. Sometimes even with the aid of ZVC, the sensor signal was not good enough to present the letter clearly. Sensor drift mainly meant that we could not reconstruct the letter that we had written. Besides this, there were several other factors to affect the performance. Firstly, the different tracks. From Figure 5.19, it can be seen that the shapes of *s* were not exactly the same, which might lead to different motion properties of each stroke. Secondly, the sensor performance also depends on the stroke - how fast it was written each time and how smooth it was. The sensors responded differently even to the same letter written by different motions. Finally, attitude might change more or less during one stroke. A fixed attitude is necessary to guarantee more accurate results because there is an initial estimate of the attitude, which will be brought into the calculation of the sensor output. Hence, we can conclude from this test that the current conditions probably cannot provide us with a consistent result from sensors when writing a single letter repeatedly. People can hardly recognize the letter from the sensor output, and even recognition software has difficulty identifying the different letters. More experiments like this will be carried out in the near future to determine whether the sensors can give consistent results for the same letter or not.

5.5 Analysis on Resolution Property of Current Vision Algorithm

5.5.1 Resolution of Current Algorithm

The vision algorithm is used to calculate 2D velocity and acceleration, including two parts: one is the matching, which gives the position information of the pen-tip from the image sequence; the other is the calculation of velocities and accelerations based on the previously obtained position information. In the first part, the minimum measurement unit can reach one pixel already, so the main factor the resolution depends on is the calculation.

Because of the high sampling rate of the camera, there must be some very small movements, which the matching may incorrectly regard as being stationary. To reduce the effect of these mistakes, a simple filter was added to deal with the position information before calculation. The concept of this filter is to average the image coordinates of five neighboring points and bring this averaged value as the coordinate of the middle point into the latter calculation. It can be briefly presented as follows:

$$x(i)=\frac{u(i-2)+u(i-1)+u(i)+u(i+1)+u(i+2)}{5} \qquad (33)$$

Hence, the minimum measurement unit is reduced by four-fifths at most, at 0.2 pixel by this filter. According to the mathematical model assumed as shown in equations (6) and (7), we can get the equation for the calculation of velocity below:

$$V_{x\_n}=\frac{K\cdot(x_{n+1}-x_n)}{\Delta t} \qquad (34)$$

Where, x_n is the averaged position in x axis of the n^{th} point by the filter; V_{x_n} is the magnitude of velocity in the x axis. As to the parameter *K*, we have obtained it for each character and illustrate them in Table 5.1, then calculate an average value for it, which is equal to 1.144e^{-4} meters per pixel. Finally, we get the minimum measurement units of displacement, velocity and acceleration, which are equal to 2.288e^{-5} meters, 4.576e^{-3} meters per second and 0.9152 meters per second squared respectively.

5.5.2 Tests with Various Filters

Besides the currently used filter of 5 points averaged, others methods, including without a filter, were also tested with different numbers of point averaged. Eventually, the filter is very effective to smooth the curves of velocities, especially for the accelerations and the higher quantity of points we averaged, the better resolution and smoother curves we got. Figure 5.22 and Figure 5.23 illustrate this property from acceleration and velocity curves from the letter *s0*. In addition, we found that almost the same shape could be reconstructed from different cases (see Figure 5.24), which means all the velocities and accelerations were reliable. However, a filter with a high quantity of points averaged may lead to a loss of true motion information, a fierce instant movement for example, and does not work that significantly in smoothing curves when it already arrived at some level before. Even though a satisfactory reference has been able to be achieved, we will improve the algorithm to be better in accordance with our practical demands if it is confirmed that the current filter is not sufficient for future application.

Table 5.1 Values of proportional parameter *K*

letter	parameter *K* (m/pixel)
a	1.13E-04
b	1.14E-04
c	1.13E-04
d	1.13E-04
e	1.14E-04
f	1.14E-04
g	1.13E-04
h	1.13E-04
i	1.15E-04
j	1.14E-04
k	1.13E-04
l	1.13E-04
m	1.13E-04
n	1.17E-04
o	1.14E-04
p	1.13E-04
q	1.15E-04
r	1.13E-04
s	1.17E-04
t	1.13E-04
u	1.18E-04
v	1.14E-04
w	1.19E-04
x	1.14E-04
y	1.17E-04
z	1.16E-04

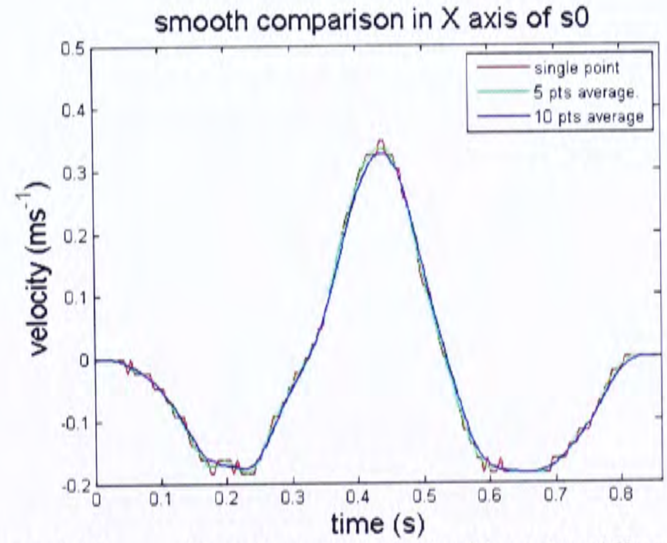


Figure 5.22 Smooth curves of velocities with different filters

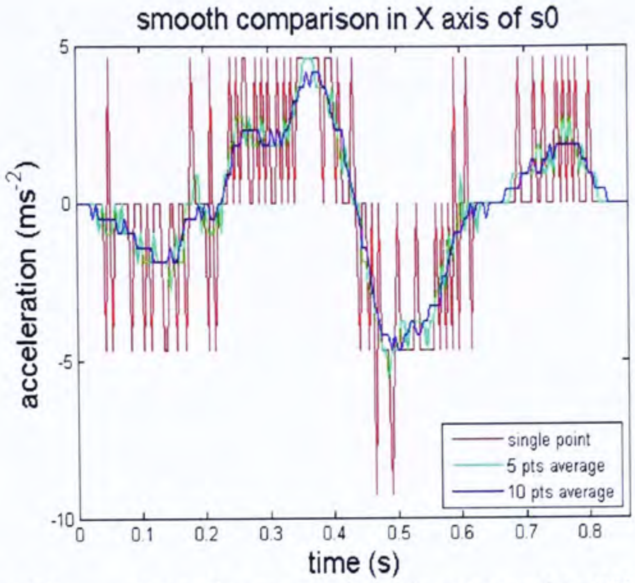


Figure 5.23 Smooth curves of accelerations with different filters

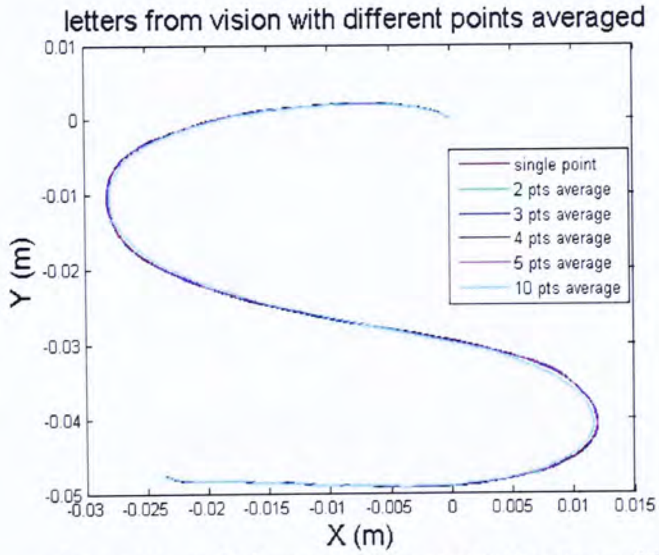


Figure 5.24 Reconstructed characters from optical system with different filters

5.6 Calculation of Static Attitude

Figure 5.25 shows a pair of pictures taken by the left and right cameras respectively with the method as shown in Figure 4.2, and these are selected to realize a 3D reconstruction. In this procedure, we took pictures of the template, using both cameras at the same time, with five different orientations of the board. That is because several pairs provide more information for camera calibration, which can improve the accuracy for both the intrinsic and extrinsic parameters of the cameras. The target in this experiment is the black pen which was stuck closely on to the template plane. The coordinates of two top points on the pen, which were described in the world reference frame, are calculated after stereo calibration. The frame was defined by us as shown in Figure 5.25, of which the XY plane was the same as the template plane. The coordinates in the image reference frame of the points to be reconstructed are also shown in it. Also after the points were reconstructed, three angles could be obtained between the line made up of the two points and the three axes of the world frame. Since the length of the pen is known to be 140mm, so the distance between the two reconstructed points could be calculated and regarded as a reference for the reconstruction.

From Figure 5.26 and Table 5.2, information about both the attitude and position described in the world frame are presented. For attitude result, there was approximately 2 degrees offset from the physical measurement. Measurement and pixel point selection errors might lead to this offset. Since the error of length comparison was only 1.6%, the result of position calibration was good and acceptable. Eventually, it provides us with an optimistic outlook to further develop the attitude calibration based on this method, even though there will be some difficulties to face in the future.

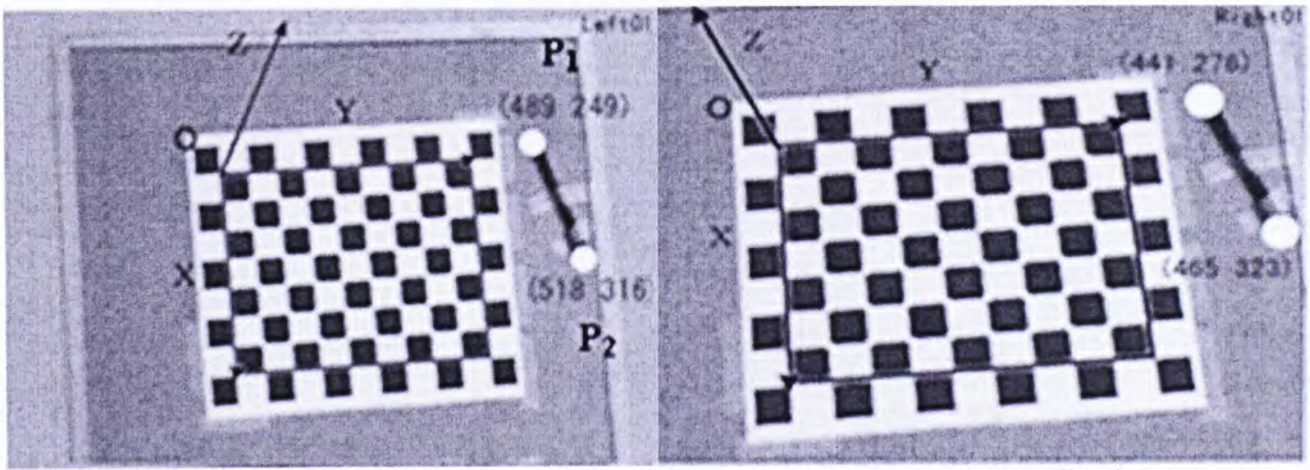


Figure 5.25 A pair of pictures for 3D reconstruction after camera calibration

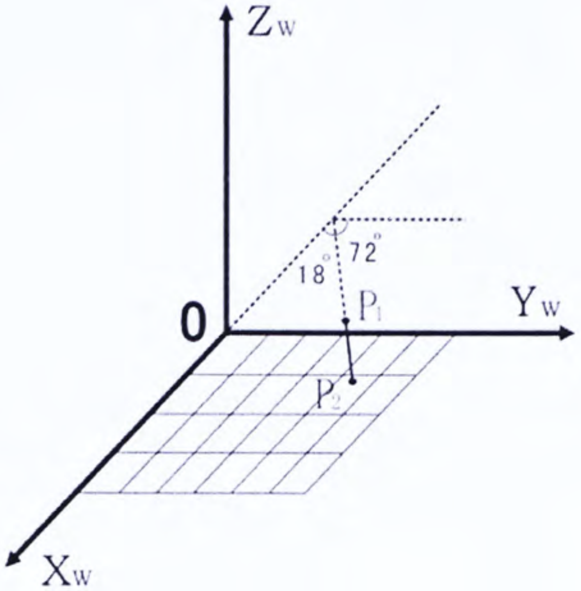


Figure 5.26 Experimental description by projection relationship

Table 5.2 Results of attitude and position based on multiple camera calibration with the parameters of the left camera

<table>
<tr><th></th><th colspan="2">Physical Measurement Results</th><th colspan="2">Calculated Results from 3D Reconstruction</th></tr>
<tr><td>Angle between X axis (°)</td><td colspan="2">18</td><td colspan="2">16.58</td></tr>
<tr><td>Angle between Y axis (°)</td><td colspan="2">72</td><td colspan="2">73.44</td></tr>
<tr><td>Angle between Z axis (°)</td><td colspan="2">90</td><td colspan="2">89.23</td></tr>
<tr><td>Length (<i>mm</i>)</td><td colspan="2">140</td><td colspan="2">137.76</td></tr>
<tr><td></td><td>X</td><td colspan="2">Y</td><td>Z</td></tr>
<tr><td>Position of Point1 (<i>mm</i>)</td><td>-15.04</td><td colspan="2">335.15</td><td>10.62</td></tr>
<tr><td>Position of Point2 (<i>mm</i>)</td><td>116.99</td><td colspan="2">374.41</td><td>12.47</td></tr>
</table>

Chapter 6

Future Work

6.1 Another Multiple Tests of Letter *k*

Assuming that writing a single letter for several times can give the same output signal from the accelerometers, a pointed algorithm can be easily developed to complete the recognition task for various letters. Besides the sample letter introduced in Section 5.4, another set of tests on letter *k* was also completed. In this experiment, the letter was written five times and the durations were about 1.375, 1.450, 1.475, 1.750 and 1.425 seconds each time respectively. Figure 6.1 illustrates the exact written letters, which we obtained from the video sequences and matching algorithm. The results based on the sensor's raw data are shown in Figure 6.2, from which we can say that it is hardly to be used for recognition from the sensor's raw data directly. Only the trends of the curves were similar to each other. A different situation happened after the sensor data was processed with ZVC. Although we still cannot recognize the letters as *k*, the orientations and shapes of the curves are approximately consistent (see Figure 6.3).

Then we may conclude that it is very hopeful that an algorithm can be successfully developed for the recognition task based on the sensor output after processing the sensor raw data with ZVC. More tests are needed to prove this conclusion in the future, especially for the case of those more complicated characters.

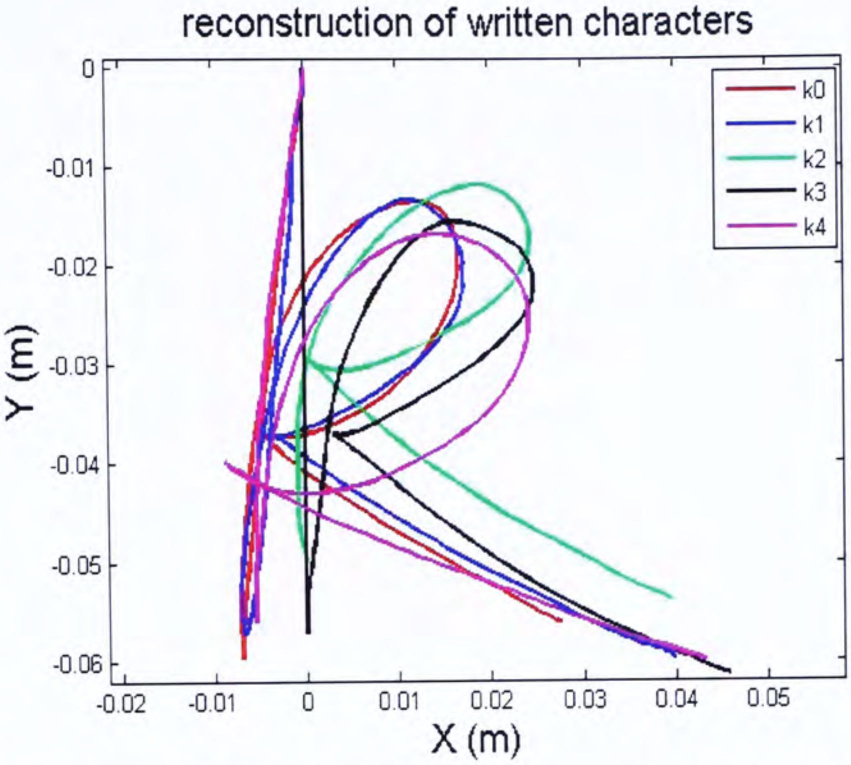


Figure 6.1 Written letter *k* from optical system and matching algorithm

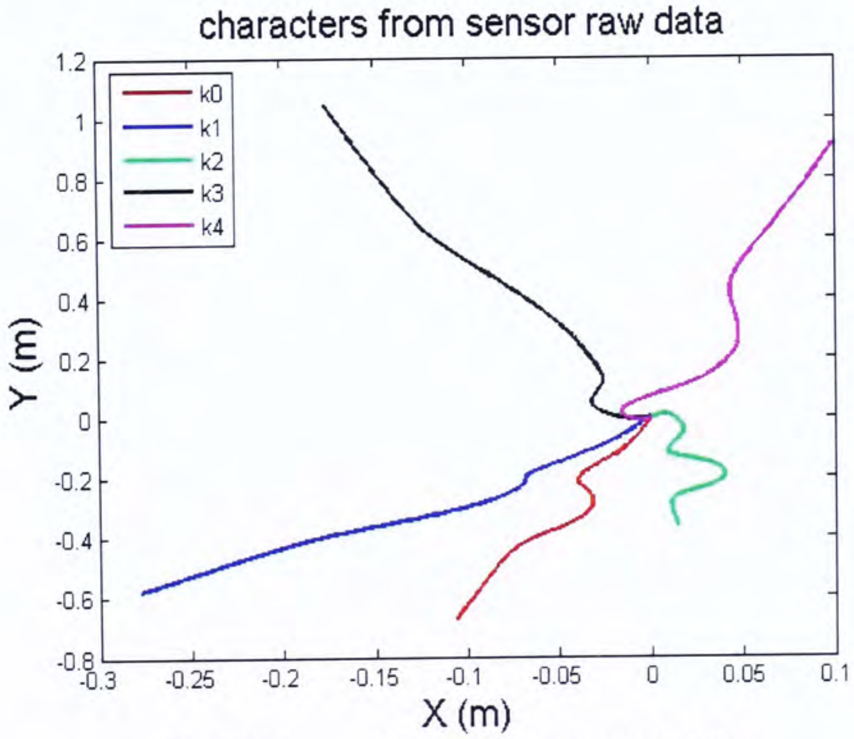


Figure 6.2 Letter *k* from sensor raw data

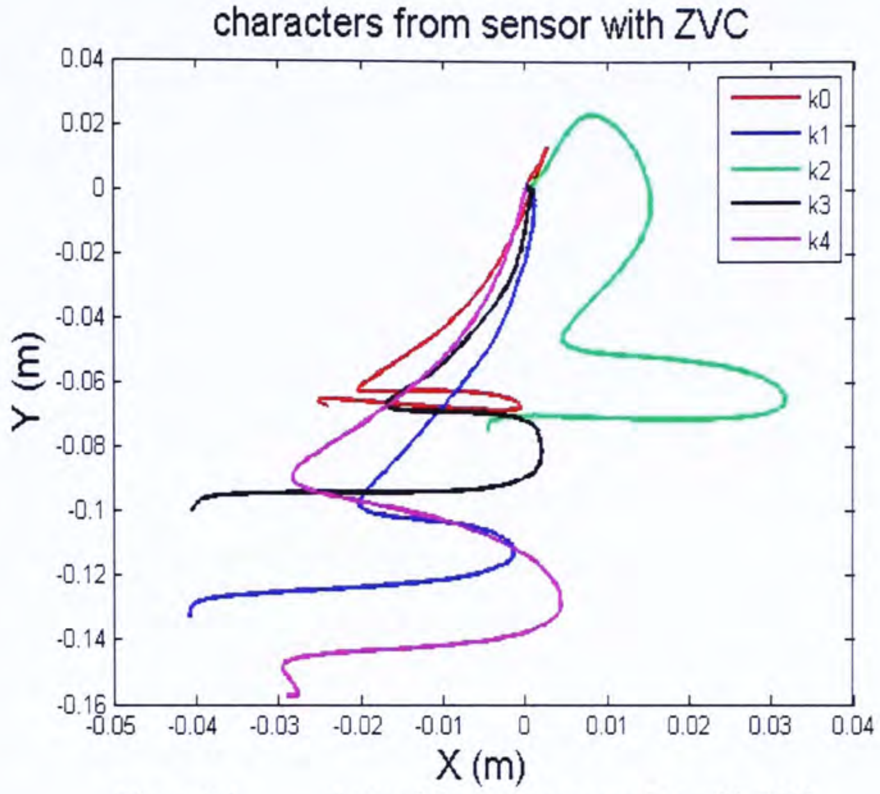


Figure 6.3 Letter *k* from sensor raw data with ZVC

6.2 Letter Recognition Based on Neural Networks Classification

According to the results discussed in Section 5.4 and 6.1, several techniques are potentially useful in realizing real-time letter recognition using MEMS motion sensors, for example, the classification based on Neural Networks. Actually, our group has done some similar experiments to classify different letters written by another μIMU through a Hidden Markov Model (HMM) from Markov Chains, which is a doubly embedded stochastic process with an underlying stochastic process that is not observable, and can only be observed through another set of stochastic processes that produce the sequence of observations.

We have also performed a set of experiments using a sensing system that is much simpler than the UDWI and which consists of 2D accelerometers and 1D gyroscope as shown in Figure 6.4. Then we wrote eleven different English letters and each of them was written fifty times, i.e., we obtained five hundred and fifty training samples in total. Figure 6.5 illustrates a window-width clip of letter *A*. Since the width of each letter was different, we had to compress the data with the method of Fast Fourier Transform (FFT). After FFT for each window, the first twenty numbers were used and it gave a 60x1 vector as one independent training sample. The final FFT result of letter A is shown in Figure 6.6.

In this way, we tried to separate all these eleven different letters with their training information from each other. The final result is shown in Table 6.1. In the first case, all of this sensing system's output was adopted with their first twenty numbers in each window. Here, certain letters could be recognized from the group with high accuracy while others could not. In the second case, the information from the gyroscope was omitted and the result became much better than the previous one. In the last case, the result reached its peak when we adopted all of the sensing system's output but with the first ten numbers of each window after compressed through FFT. As shown, the accuracy could be better than ninety percent. Although more tests are necessary to verify exactly how well this method works in this application, the preliminary result is so encouraging that we regard this method as a potential technique to enable MEMS-based letter recognition sensing system.

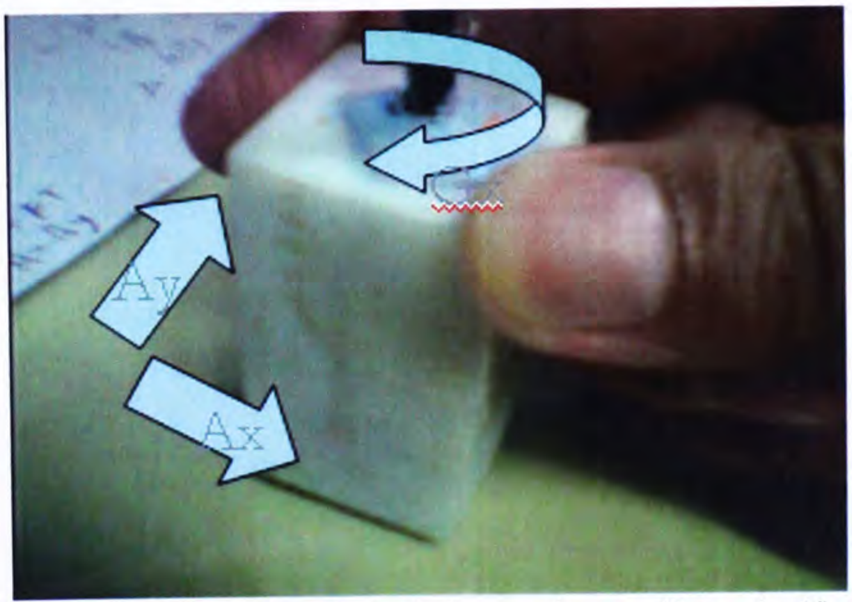


Figure 6.4 The writing device in the experiments of letter classification

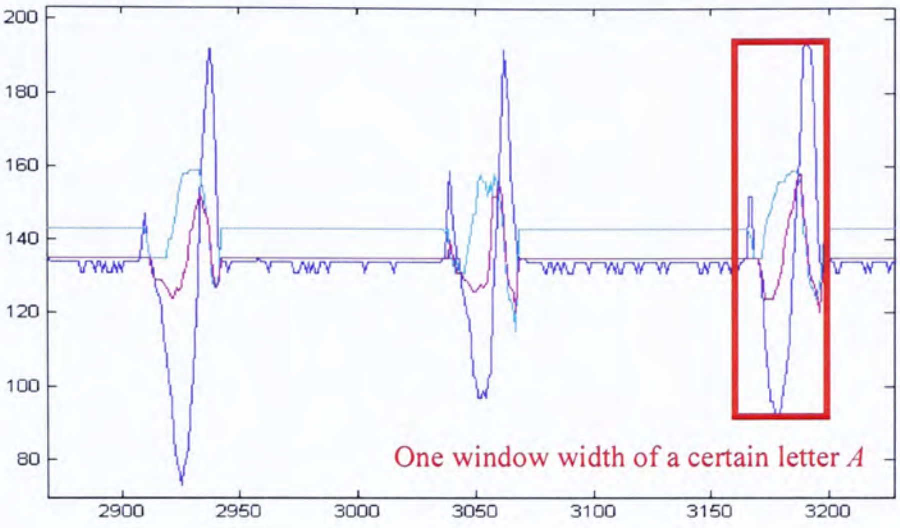


Figure 6.5 The window-width clip of writing letter *A*

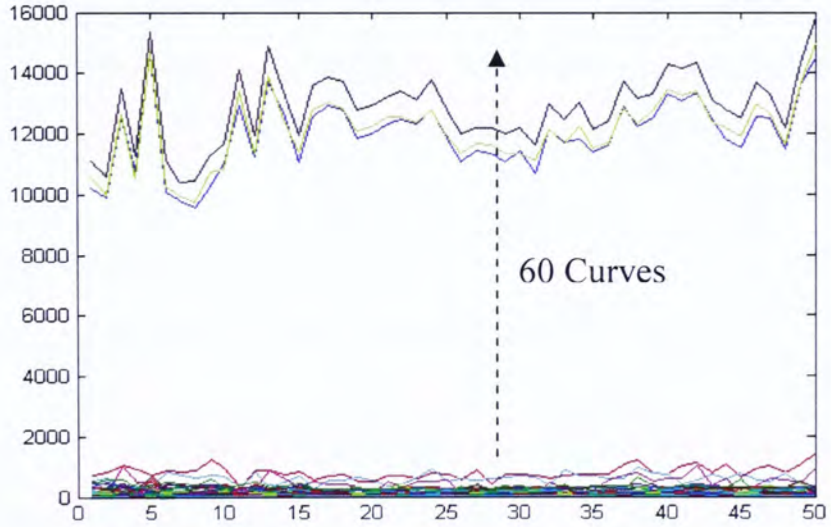


Figure 6.6 FFT results of letter A

Table 6.1 Result of letter recognition based on HMM

Letter	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*M*	*N*	*S*	*W*
Case1	49	13	46	9	50	32	26	40	32	48	46
Case2	46	43	49	45	50	48	3	45	12	30	43
Case3	43	47	48	42	50	45	45	47	32	41	47

Chapter 7

Conclusion

A Wireless Digital Writing Instrument based on MAG-μIMU has been developed with three-dimensional accelerometers, gyroscopes and magnetometers with strap-down installation. Through Bluetooth wireless connection, the motion data are transmitted to a computer with 200Hz sampling rate for real-time processing of attitude determination and position tracking. However, because of the sensors' intrinsic bias and random noise such as circuit thermal noise, a calibration system that provides good reference measurement parameters must be used to compare with the output of the MAG-μIMU sensors.

7.1 Calibration of MAG-μIMU Sensors

This thesis presents a complete design of an optical tracking system for calibrating the accelerometers applied into our UDWI. This calibration system consists of two parts: one is the 2D motion calibration for the accelerometers; the other one is the attitude calibration for the gyroscopes and the magnetometers. When the two parts combined together, all the nine-dimensional output of our MAG-μIMU sensors can be rectified according to the relatively more accurate motion information obtained from the optical tracking system. Even though all the functions have not yet been realized, current experimental results are still proved feasible and have potential enough to bring us encouragement.

7.2 Calibration of Accelerometers

In this task, a transparent table with adjustable height was built for writing. In addition, a high speed camera supplies us with a technical support of motion estimation with video compression aid. Based on the image sequence captured by the camera when UDWI is writing on the table, our matching algorithm can help us to analyze the motion properties of the pen-tip where the accelerometers are attached. Through the comparisons between the optical results and sensors output, it is found that the sensor drift effect occurs when we integrate the accelerations to the velocities and displacements. When the ZVC algorithm is brought into the integrations, it further proves that the results from the optical tracking technique are more accurate and reliable as good references. All the English letters in lowercase have been written in this way and some of them were repeated several times. It is believed that these references will be very helpful to overcome the sensor drift better in the future.

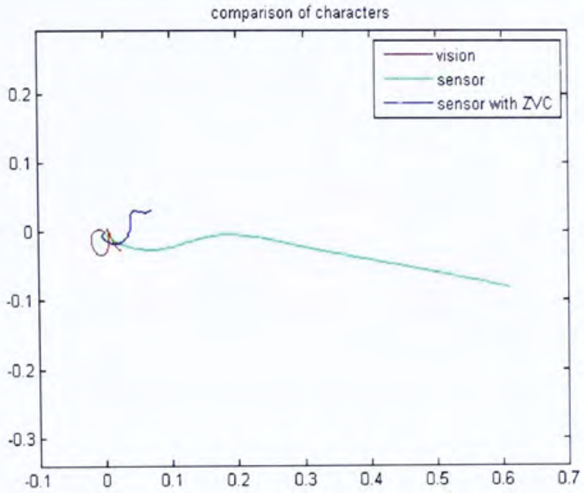
7.3 Calibration of Attitude

Three-dimensional attitude calibration is realized through camera calibration technique, especially the multiple camera calibration. We can ascertain all the necessary parameters about the cameras through a toolbox based on MATLAB, which provides us with the functions of camera calibration and stereo calibration. Based on these parameters, including the intrinsic and extrinsic parameters and relative space relationship of the cameras, a 3D point in the space is able to be reconstructed with its coordinates known from our own algorithm. Finally, a static attitude can be ascertained with this multiple camera system.

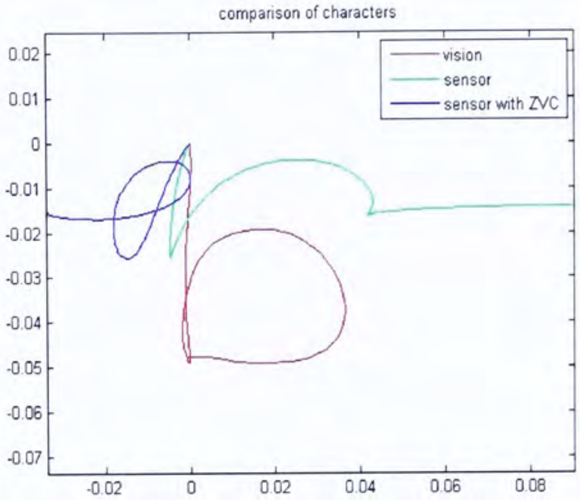
7.4 Future Work

From the English letter-writing experiments, we have proved that the most dependable motion information such as displacement, velocity and acceleration, can be obtained through the optical tracking system. Hence, we can now compare the MEMS accelerometer output results with our optical tracking system output. However, although the basic function of 3D space reconstruction using the MEMS sensors was realized in our lab, the gyroscopes and magnetometers were not calibrated yet. The tasks of space position and attitude calculation will be carried out in the future by other researchers to completely finish the calibration of the entire nine-dimensional output from the MAG-μIMU.

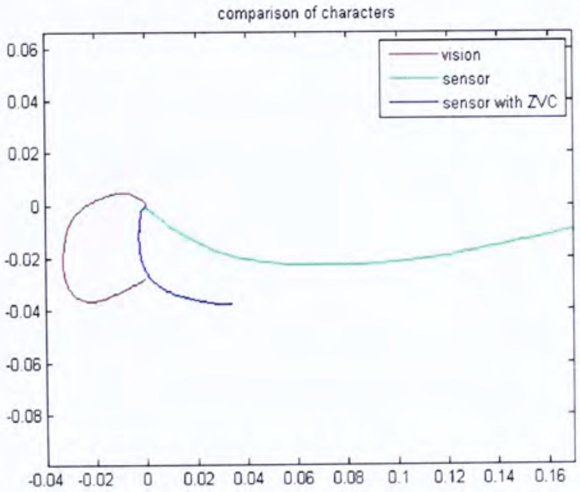
Appendix A
The Experimental Results of Writing English Letters



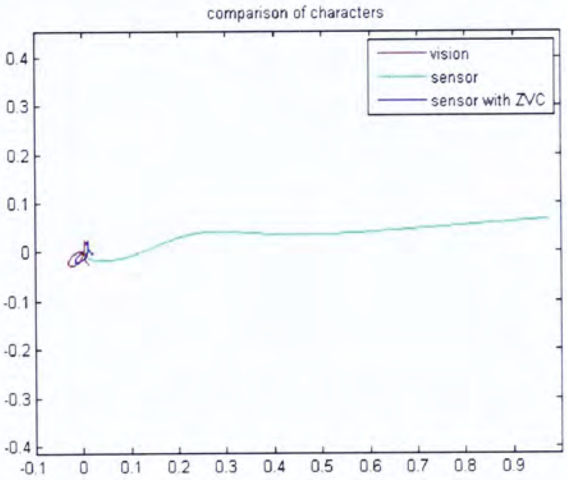
Result Figure 1 Letter *a*



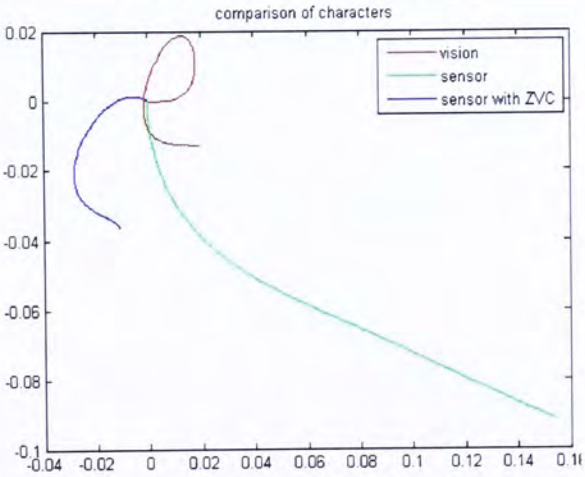
Result Figure 2 Letter *b*



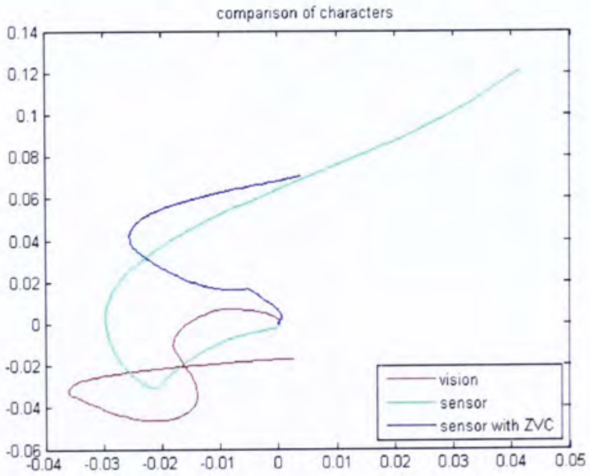
Result Figure 3 Letter *c*



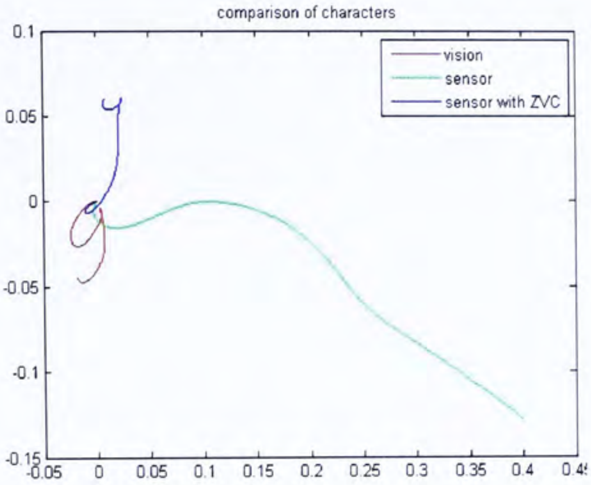
Result Figure 4 Letter *d*



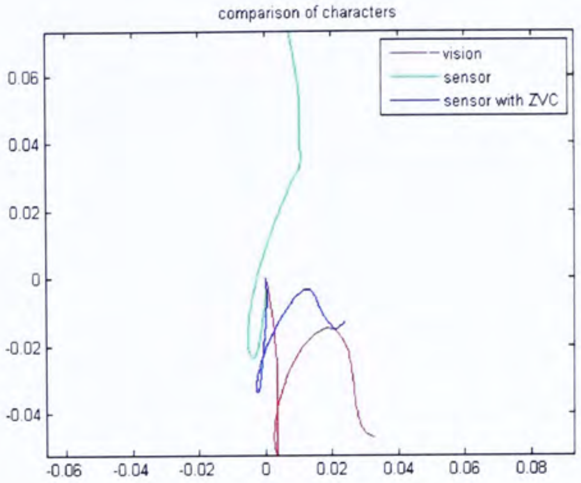
Result Figure 5 Letter *d*



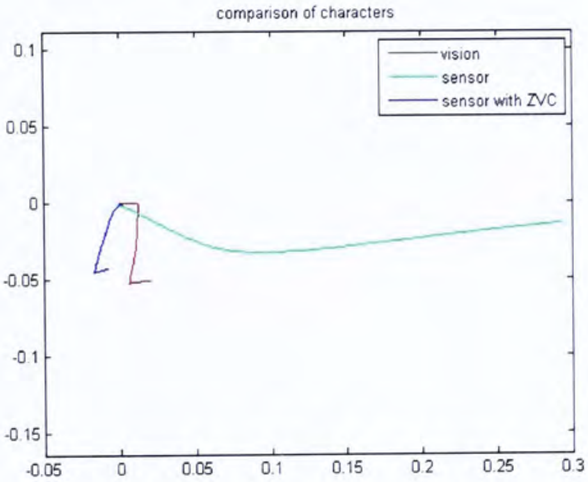
Result Figure 6 Letter *f*



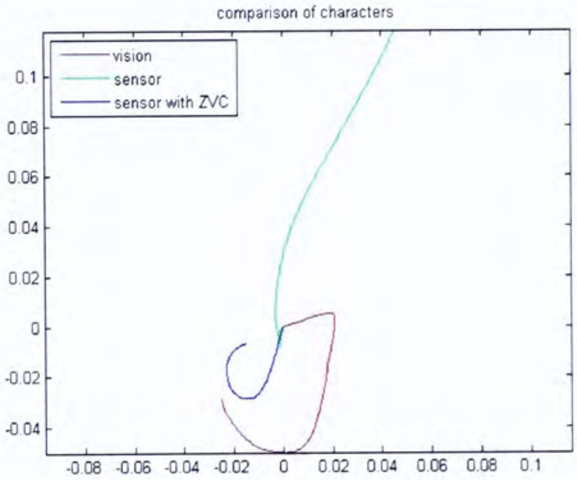
Result Figure 7 Letter *g*



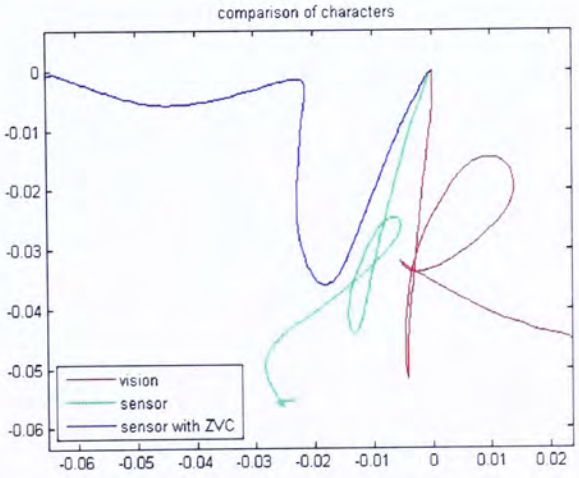
Result Figure 8 Letter *h*



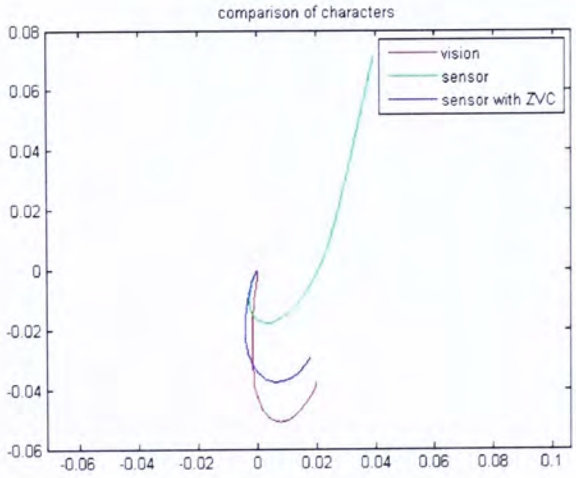
Result Figure 9 Letter *I*



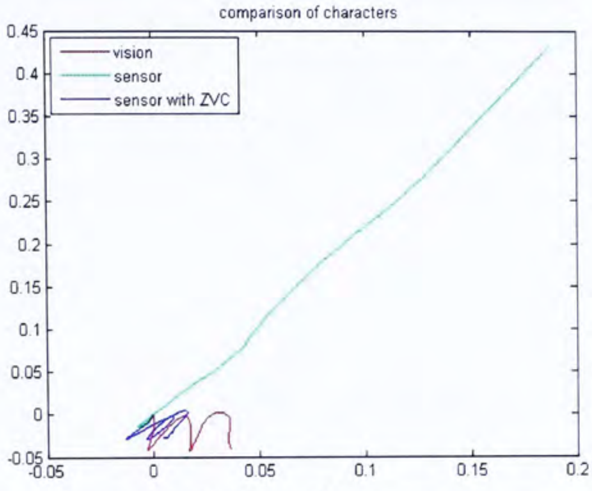
Result Figure 10 Letter *J*



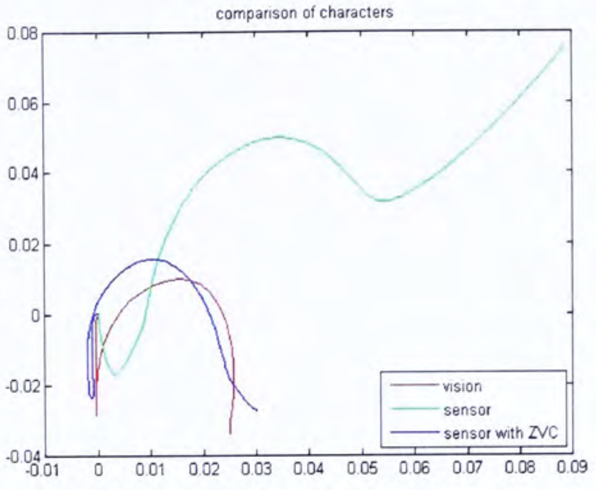
Result Figure 11 Letter *k*



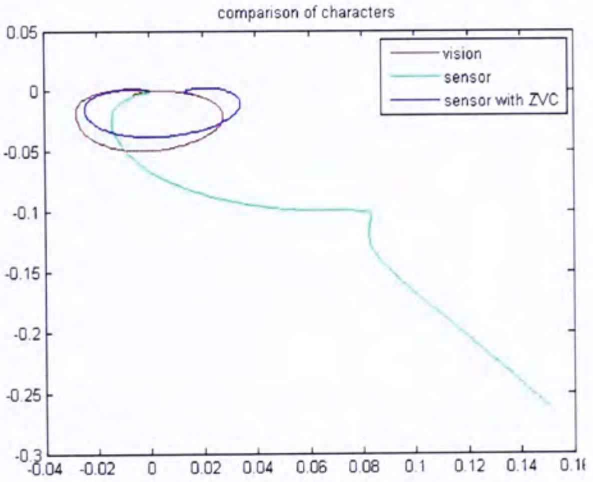
Result Figure 12 Letter *l*



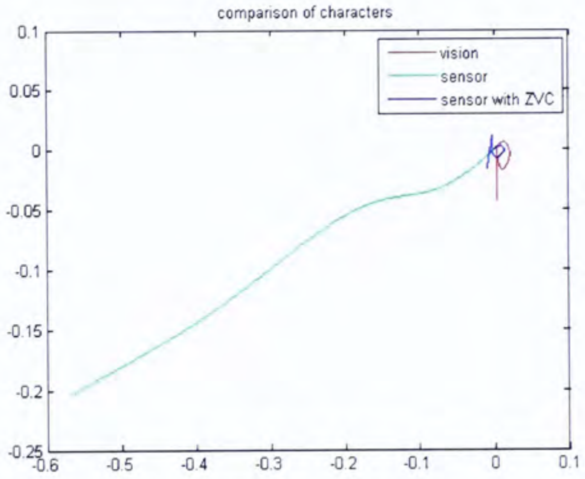
Result Figure 13 Letter *m*



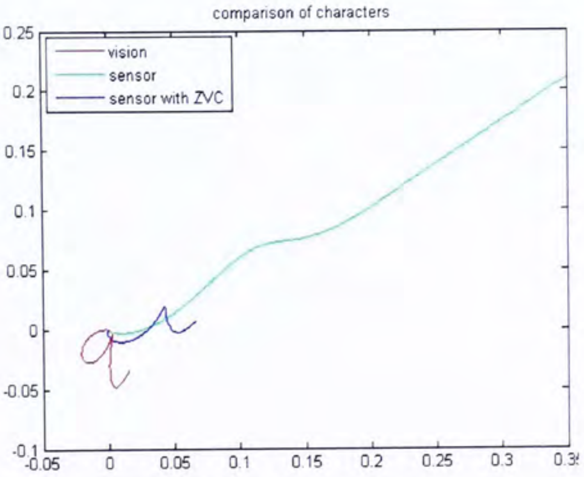
Result Figure 14 Letter *n*



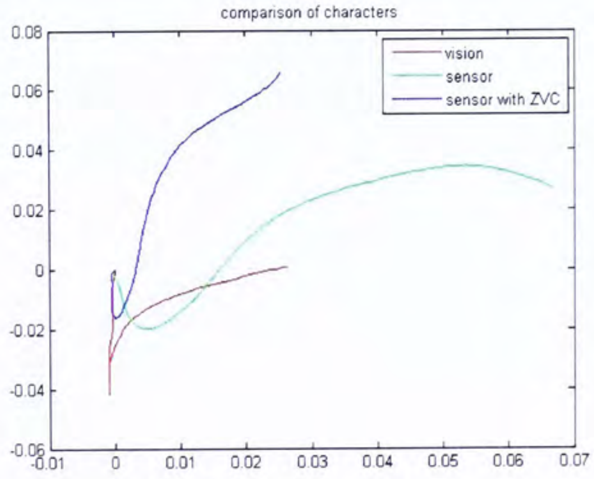
Result Figure 15 Letter *o*



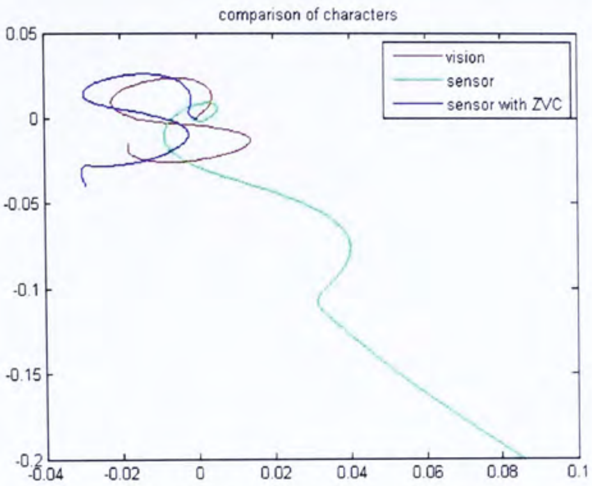
Result Figure 16 Letter *p*



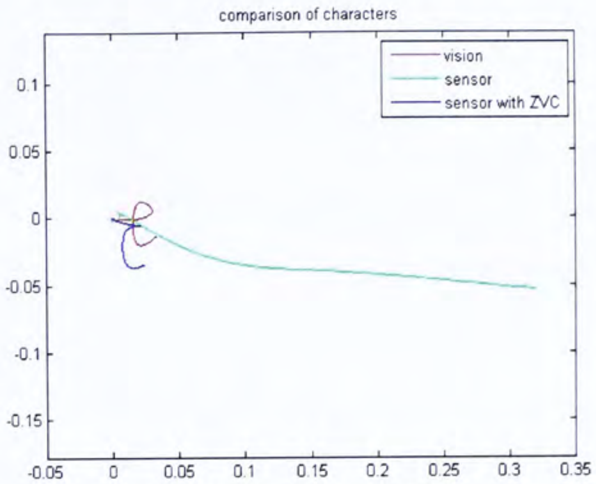
Result Figure 17 Letter *q*



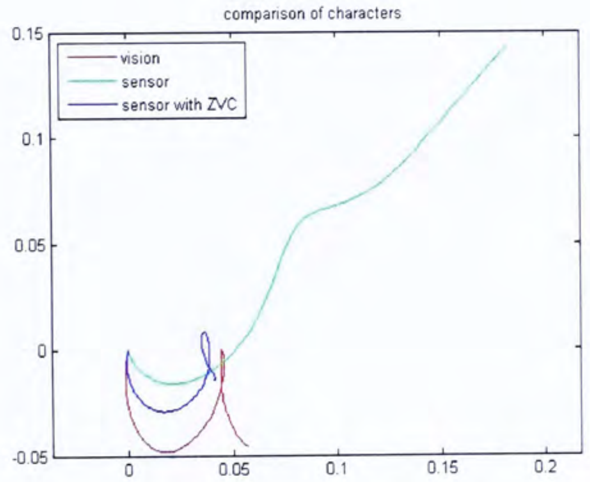
Result Figure 18 Letter *r*



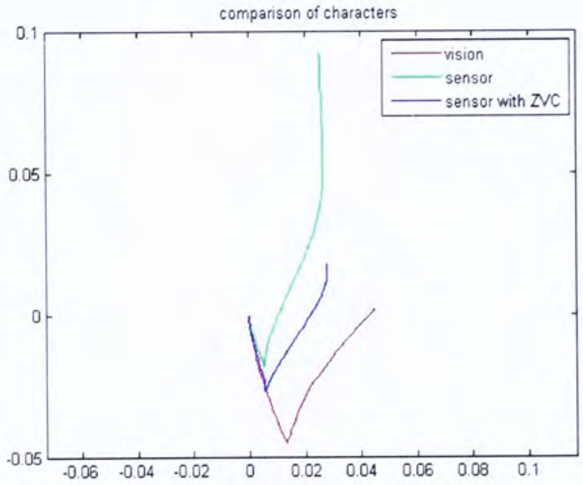
Result Figure 19 Letter *s*



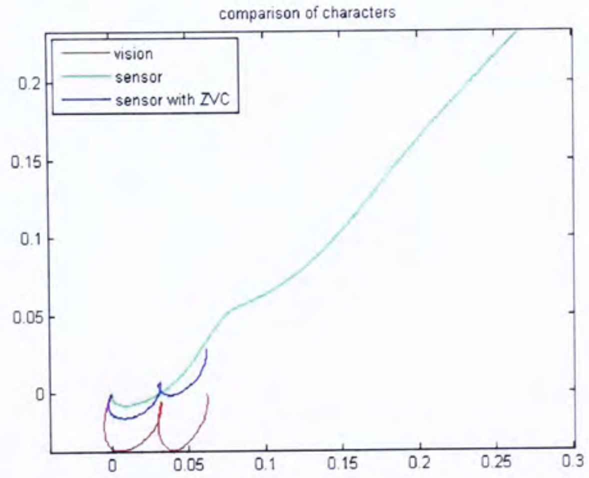
Result Figure 20 Letter *t*



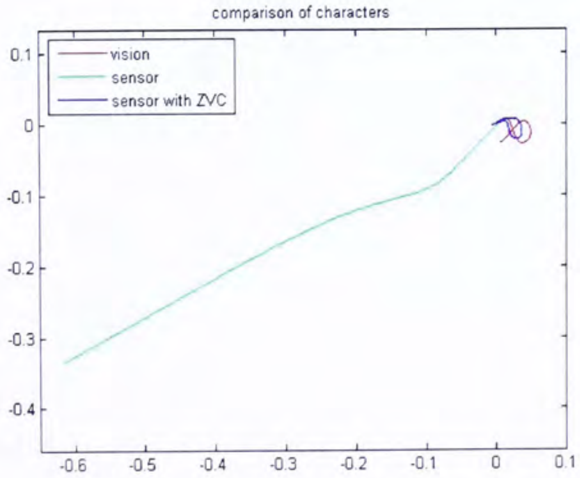
Result Figure 21 Letter *u*



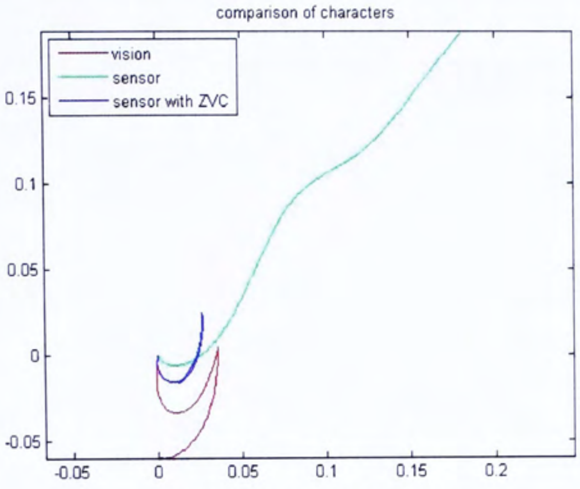
Result Figure 22 Letter *v*



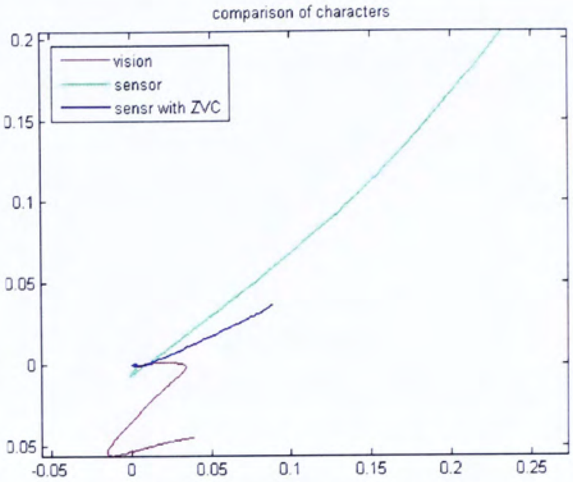
Result Figure 23 Letter *w*



Result Figure 24 Letter *x*



Result Figure 25 Letter *y*



Result Figure 26 Letter *z*

Bibliography

[1] J.L. Marins, Y. Xiaoping, E.R. Bachmann, R.B. McGhee, M.J. Zyda, "An extended Kalman filter for quaternion-based orientation estimation using MARG sensors," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems 2001*, Vol.4, pp.2003-2011.

[2] E. Foxlin, "Inertial hear-tracker sensor fusion by a complementary separate-bias Kalman filter," *Proc. of the IEEE Virtual Reality Annual International Symposium*, March 30-April 3, 1996, pp.185-194.

[3] Sagem Avionics, Inc. APIRS F200-AHRS Brochure ,2006. URL: http://www.sagemavionics.com/Products/Brochures/APIRS-F200.pdf.

[4] Watson Industries, Inc. AHRS-E304 Manual, 2006. URL: http://www.watson-gyro.com/files/attitude_reference_AHRS-E304_spec.pdf.

[5] Roger C. Hayward, Demoz Gebre-Egziabher, Matthew Schwall, J. David Powell and John Wilson. "Inertially Aided GPS Based Attitude Heading Reference System (AHRS) for General Aviation," Proceedings of ION GPS '97, Kansas City, MO, September, 1997.

[6] Alison K. Brown, "Test Results of a GPS/Inertial Navigation System Using a Low Cost MEMS IMU", 11th Annual Saint Petersburg International Conference on Integrated Navigation System, Saint Petersburg, Russia, May, 2004.

[7] Eun-Seok Choi, W. Chang, W. C. Bang, J. Yang, S. J. Cho, J. K. Oh. J. K. Cho, and D.Y. Kim, "Development of the Gyro-free Handwriting Input Device based on Inertial Navigation System (INS) Theory," Proceedings of SICE Annual Conference '2004, SICE, 2004, pp. 1176-1181, vol. 2.

[8] G. L. Zhang, G. Y. Shi, Y. L. Luo, H. Wong, Wen J. Li, Philip H. W. Leong, and Ming Yiu Wong, "Towards an Ubiquitous Wireless Digital Writing Instrument Using MEMS Motion Sensing Technology," *Proceedings AIM '2005*, IEEE/ASME, 2005, pp. 795-800.

[9] Yilun Luo, Chi Chiu Tsang, Guanglie Zhang, Zhuxin Dong, Guangyi Shi, Sze Yin Kwok, Wen J. Li, Philip H.W. Leong, Ming Yiu Wong, "An Attitude Compensation Technique for a MEMS Motion Sensor Based Digital Writing Instrument", *Proceedings of 2006 IEEE International Conference on Nano/Micro Engineered and Molecular Systems, 2006.*

[10] Chi Chiu Tsang, Gary Chun Tak Chow, Guanglie Zhang, Yilun Luo, Zhuxin Dong, Guangyi Shi, Sze Yin Kwok, Heidi Y. Y. Wong, Philip H. W. Leong, Wen

J. Li and Ming Yiu Wong, "A Novel Real-Time Error Compensation Methodology for μIMU-based Digital Writing Instrument", *Proceedings of 2006 IEEE International Conference on Robotics and Biomimetics(IEEE-ROBIO 2006)*, Kunming, China, 17-20 December 2006.

[11] Analog Devices, Inc. Low-Cost Ultracompact ±2g Dual-Axis Accelerometer ADXL311 Data Sheet (Rev. B, 01/2005), 2005. URL: http://www.analog.com/UploadedFiles/Data_Sheets/243920868ADXL311_B.pdf.

[12] Murata Manufacturing Co., Ltd. GYROSTAR Piezoelectric Vibrating Gyroscopes Data Sheet (No. S42E, 06/2006), 2006. URL:http://www.murata.com/catalog/s42e.pdf.

[13] Koninklijke Philips Electronics N. V. Magnetic Field Sensor KMZ51 Data Sheet (06/2000),2006.URL:http://www.nxp.com/acrobat_download/datasheets/KMZ51_3.pdf.

[14] Koninklijke Philips Electronics N. V. Magnetic Field Sensor KMZ52 Data Sheet (06/2000), 2006. URL: http://www.nxp.com/acrobat_download/datasheets/KMZ52_1.pdf.

[15] Honeywell International, Set/Reset Function for Magnetic Sensors (08/02), 2003. URL: http://www.ssec.honeywell.com/magnetic/datasheets/an213.pdf.

[16] Atmel Corporation. 8-bit AVR Microcontroller with 32 Bytes In-System Programmable Flash ATmega32(L) Data Sheet (Rev. I, 04/06), 2006. URL: http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf.

[17] Richard H. Barnett, Sarah Cox, Larry O'Cull, *Embedded C Programming and the Atmel AVR*, Delmar, Clifton Park, NY, USA, 2003.

[18] BlueRadios, Inc. Class 1, Class 2 and 3 BR-C30 ver 1.2 Module Specification, 2006. URL: http://www.blueradios.com/BR-C30.pdf.

[19] IVT Corporation, BlueSoleil 2.3 Commercial Version Release Note (Ver. 2.3, 07/06), 2006. URL: http://www.bluesoleil.com/Download/files/IVT_BlueSoleil_%28Standard%29_Release_Note.pdf.

[20] Future Technology Devices International Ltd. FT232BM USB UART (USB-Serial) I.C. Data Sheet (Ver. 1.8, 2005), 2006. URL: http://www.ftdichip.com/Documents/DataSheets/ds232b18.pdf.

[21] D. H. Tittertoin, J. L. Weston, *Strapdown Inertial Navigation Technology, 2nd Edition*, AIAA, Reston, USA, 2004.

[22] R.M. Rogers, *Applied Mathmatics In Integrated Navigation Systems, Second Edition.* AIAA Education series, 2003.

[23] A. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. American Institute of Aeronautics and Astronautics, 1997.

[24] J.C. Hung, J.R. Thacher and H.V. White, "Calibration of Accelerometer Triad of An IMU with Drifting Z-Accelerometer Bias", in *Proc.NAECON 1989, IEEE Aerospace and Electronics Conference*, 22-26 May, 1989, vol.1,pp.153-158.

[25] The Model Shop INC., URL: http://www.modalshop.com/techlibrary/Shock%20and%20Vibration%20Calibration%20_0603peres.pdf

[26] A. Kim and M.F. Golnaraghi, "Initial Calibration of An Inertial Measurement Unit Using An Optical Position Tracking System", *in Proc. PLANS 2004, IEEE Position Location and Navigation Symposium*, 26-29 April, 2004, pp.96-101.

[27] Xuan Jing and Lap-Pui Chau, "An Efficient Three-Step Search Algorithm for Block Motion Estimation," IEEE Trans. on Multimedia, vol. 6, no: 3, June, 2004.

[28] Renxiang Li, Bing Zeng, and Ming L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, no: 4, pp. 438-442, Aug. 1994.

[29] Lai-Man Po and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, no: 3, June, 1996.

[30] P. Baglietto, M. Maresca, A. Migliaro and M. Migliardi, "Parallel Implementation of the Full Search Block Matching Algorithm for Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., pp. 182-192, 1995.

[31] Klaus Strobl, Wolfgang Sepp, Stefan Fuchs, Christian Parades and Klaus Arbter, "Camera Calibration Toolbox for Matlab". URL:http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

[32] Homepage of pco.imaging corparation. URL: http://www.pco.de

[33] Songde Ma, Zhengyou Zhang, Computer Vision— Theorem of Computation and Basic Algorithm, Science Press, China, 1998.

[34] K. B. Atkison Ed., Developments in Close Range Photogrammetry, London, 1980.

[35] J. Weng, P. Cohen and M. Herniou, Calibration of Stereo Camearas Using a Non-linear Distortion Model, *Proc. International Conference on Pattern Recognition*, pp.246-253, 1990.

[36] R. Y. Tsai, An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.364-374, 1986.

[37] O. D. Faugeras and G. Toscani, Camera Calibration for 3D Computer Vision, *Proc. of International Workshop on Industrial Application of Machine Vision and Machine Intellegence*, pp.240-247, Japan, 1987.

[38] Zhengyou Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol.1, no.pp.666-673 vol.1, 1999.

[39] Heikkila, J.; Silven, O., "A four-step camera calibration procedure with implicit image correction," *Proceedings 1997 IEEE Computer Society Conference*, vol., no.pp.1106-1112, 17-19 Jun 1997.

[40] J. More. "The Levenberg-Marquardt Algorithm: Implementation and Theory," G. A. Watson, editor, *Numerical Analysis*, Lecture Notes in Mathematics 630, Springer-Verlag, 1977.

[41] Zhuxin Dong, Guanglie Zhang, Yilun Luo, Chi Chiu Tsang, Guangyi Shi, Sze Yin Kwok, Wen J. Li, Philip H.W. Leong and Ming Yiu Wong, "A Calibration Method for MEMS Inertial Sensors Based on Optical Tracking", *Proceedings of 2007 IEEE International Conference on Nano/Micro Engineered and Molecular Systems (IEEE-NEMS 2007)*, Bangkok, Thailand, 16-19 January 2007.

[42] Zhengyou Zhang, "Camera Calibration with One-Dimensional Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, no: 7, July 2004.

[43] Fei-Yue Wang, "An Efficient Coordinate Frame Calibration Method for 3-D Measurement by Multiple Cameras Systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 35, no: 4, November 2005.

[44] Jing Yang, Wook Chang, Won-Chul Bang, Eun-Seok Choi, Kyoung-Ho Kang, Sung-Jung Cho, and Dong-Yoon Kim. Analysis and compensation of errors in the input device based on inertial sensors. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04),* volume2, pages 790-796, 2004.

[45] Won-Chul Bang, Wook Chang, Kyeong-Ho Kang, Eun-seok Choi, Alexey Potanin, ang Dong-Yoon Kim. Self-contained spatial input device for wearable computers. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC'03),* pages 26-34, 2003.