A Calibration Method for MEMS Inertial Sensors Based on Optical Tracking

Zhuxin Dong[1], Guanglie Zhang[1], Yilun Luo[1], Chi Chiu Tsang[2], Guangyi Shi[1], Sze Yin Kwok[1]
Wen J. Li[1,*], Philip H. W. Leong[2], and Ming Yiu Wong[3]

[1]*Center for Micro and Nano Systems, The Chinese University of Hong Kong, Hong Kong SAR*
[2]*Dept. of Computer Science and Eng., The Chinese University of Hong Kong, Hong Kong SAR*
[3]*DAKA Development Ltd., Kowloon, Hong Kong SAR*

***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 the handwriting on any surface. However, because of the sensors' intrinsic biases and random noise such as circuit thermal noise, a calibration system that provides good reference measurement parameters must be used to compare the output of the MAG-μIMU sensors. We propose here a novel idea 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 Trajectory Calibration that is used to obtain linear accelerations of the UDWI in a particular frame defined by us; 2) Multiple Camera Calibration that is used for attitude calibration of the UDWI. 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 these more accurate data obtained from optical tracking.**

***Index Terms*—MEMS; Digital Writing System; MAG-μIMU; Block Matching; Multiple Camera Calibration; 3D Reconstruction**

I. Introduction

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 Inertial Measurement Unit (μIMU) (see Fig.1and Fig.2)[1]. Although both the hardware and software of the UDWI have been steadily improved over the past year [2] [3], noise signal can still exist to affect the UDWI system output. The noise may include the intrinsic drift of the sensors, misalignment of the sensors during PCB integration, and random noise, which are impossible to totally eliminate. Hence, in reality, we have to compensate the sensor drift after a hand-writing stroke is completed, 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.

Perhaps the human hand is one of the most flexible part of our anatomy and both translation and rotation could happen at any or the same time. So, 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 even if they have higher motion accuracy. Additionally, most of these motion stages can only implement small accelerations while the majority of our hand motion is made up of much larger accelerations. In other words, a flexible system is needed which can work directly without the constraints of motion stages.

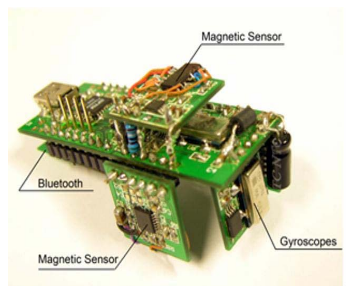


Figure 1. The Prototype of the MAG-μIMU with Bluetooth Module.

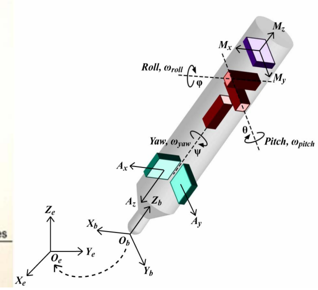


Figure 2. Reference frames for our UDWI.

Many matching algorithms and techniques have already been developed and used in the category of motion estimation with video compression aid. For example, Three Step Search (TSS) [5], Four Step Search (FSS) [6], and Parallel Full Search (PFS) [7]. Also, there are several kinds of matching criteria, e.g., Correlation Coefficient (CC), Block Distortion Measure (BDM) [4] and Mean Absolute Difference (MAD) [7]. All of these methods are matured and reliable to implement the estimation of motion vector through video sequences. In our experiment, we adopted PFS together with CC as the criteria to set up a matching system. This is a widely used method, which is most stable and convenient to realize. The disadvantage of this method is that it needs more time to calculate. However, it will not be a problem for our eventual goal because this calibration method needs not to work with a real-time system.

Besides the 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 present the attitude of the digital pen. A Camera

Contact author: **wen@mae.cuhk.edu.hk.*
This project is funded by the Hong Kong Innovation and Technology Commission (ITF-UIM-151) and by DAKA Development Ltd., Hong Kong.

Calibration Toolbox for MATLAB [8] is our fundamental tool in this part. This toolbox can provide us with single camera calibration through corners extraction, and also calibration of stereo camera systems. The helpful parameters can be guaranteed, such as camera intrinsic parameters, rotation matrix and motion vector. Based on these parameters, some other functions have been added into the toolbox by our group. Hence, a system for attitude calibration was developed.

This paper is organized as follows. In Section 2, the design of the Matching Table is introduced, including the experiment platform and software architecture. Then we will describe the design of the Attitude Calibration System in Section 3. Experimental results will be discussed in Section 4. Finally, we present conclusions and proposed future improvements in the last section.

II. MATCHING TABLE

A. Experimental Platform

The concept is that, we record a video about the motion of the pen while it is writing with a fixed orientation and distance between the camera and the calibration table. The optical axis should be vertical to the calibration plane in order to reduce the noise from the image distortion. As shown in Fig.3, the camera was located right below the transparent table, on which the pen moves to write. In this way, a video record of the motion of the pen-tip was obtained.

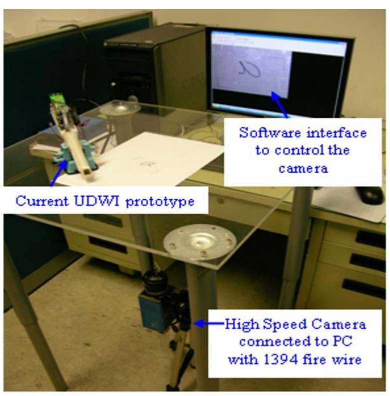


Figure 3. The experimental setup.

The video can then be divided into frames of pictures, and then frames could be analyzed to find the velocity or position of any visible point in the pictures. The sampling frequencies of our IMU sensors were set to 200*Hz*, so a high speed camera is needed to synchronize the video data to the sensors output. In fact, we adopted a camera which has a frequency range from 20 to 1000*Hz* to carry out this experiment. The time interval could reach 0.001*s* between two neighbor pictures, which is helpful for the accurate estimation of position and velocity of the pen's tip.

The trajectory was recorded on a paper with markers. We can read out the displacement both in the pixels and physical units in order to figure out the proportionality coefficients. Hence, it is convenient to transform a vector in pixel into millimeter units.

B. Software Algorithm

Fig.4 presents one of a handwriting motion sequence. After transformed into grayscale format, every pixel can be distinguished from its gray value ranged from 0 to 255. In the program, a template was defined in a rectangular area including the pen-tip according to its position. The upper-left corner's position was recorded as the position of the template. Then the calculation of the sum of gray values is 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.

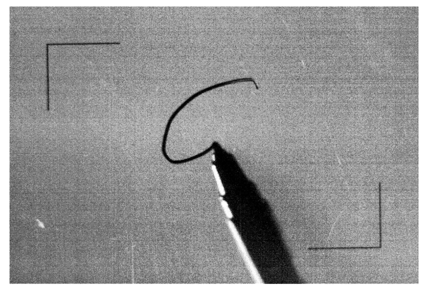


Figure 4. A picture of the writing sequence.

Then the whole picture was searched block by block, i.e.,

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

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 were calculated all over the picture.

The classical theory of finding a correlation coefficient could be presented as the following:

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

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]. In the algorithm, the two variables are the sum of gray values of the pixels in the template and search block.

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

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

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

Comparisons were done among these ρ and the block which has the highest ρ was determined. We considered this block as the right place where the pen-tip moved to and we described the location of the block also with its upper-left corner. Then the estimation about the motion vector could 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 pixels distance and physical distance will be a constant.

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

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

where u_f and v_f are the pixel coordinates results in the final found block (see Fig.5); u_s and v_s represent the position of the source block; Δt depends on the sampling frequency of the camera, i.e., $\Delta t = 1/f$. Eq.(7) above is guaranteed only if the motion plane is absolutely horizontal. Otherwise, the gravity *g* will take effects on the results of the 2D calculated accelerations.

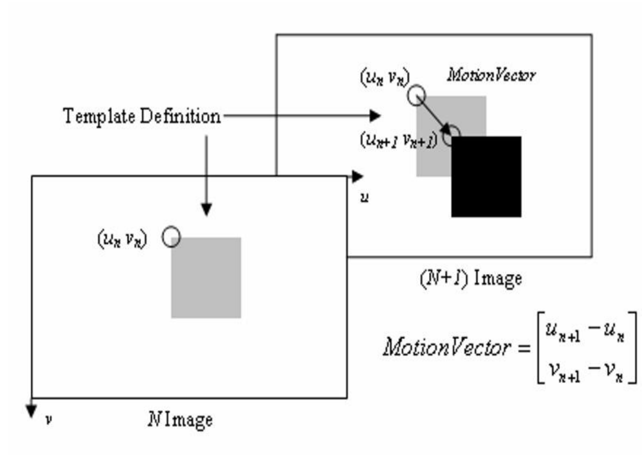


Figure 5. Description of motion vector estimation.

III. ATTITUDE CALIBRATION

Multiple Cameras Calibration [9] [10] has been applied to 3D measurements for years, so there exist many mature theories, successful experiments and dependable results. After matching, three dimensional accelerations can be guaranteed 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 Body Reference Frame (BRF), 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 realized when an attitude calibration of the pen is completed. In addition, it is also 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.

Camera calibration [11] plays an important role in attitude calibration, especially multiple camera calibration. Fig. 6 illustrates the definitions of each reference frame and the relationship can be presented as the following.

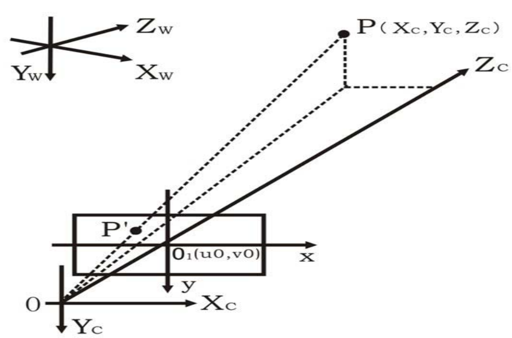


Figure 6. Illustration of the three reference frames [11].

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{8}$$

$$= \begin{bmatrix} a_x & 0 & u_0 & 0 \\ 0 & a_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M_1 M_2 X_W = M X_W$$

where M_1 M_2 are the intrinsic and extrinsic parameters' matrix; M is a 3×4 matrix called the projection matrix; dx and dy are the physical sizes per pixel along the directions of axis x and y; f is the focal length of the camera; R and t are the rotation matrix and translation vector respectively to transform between the frames; u and v are the pixel coordinates of p.

At the beginning, we do the calibration of each camera respectively, and both the intrinsic and extrinsic parameters can be guaranteed [12] [13]. The basic concept is that, firstly, print a pattern and attach it to a planar surface. Secondly, take a few 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. Next 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 [14]. After completing the iterations like above, the results become convergent and dependable.

Since one camera can only reconstruct a radial line after calibration, we need another camera to achieve an exact 3D coordinate. Based on single camera calibration, multiple camera calibration as show in Fig.7 can provide us with all the 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 Reference Frame (WRF) are defined for the cameras. According to the same WRF and the extrinsic parameters, the relative position between right and left CRF is obtained. It can be presented as:

$$X_R = R' \cdot X_L + T' \tag{9}$$

where, X_R and X_L are the coordinates vectors of each CRF; R' is the rotation matrix and T' is the translation vector. Then we can freely transform any point described in right CRF into the left one, or from left to right, according to this equation.

In the Camera Calibration Toolbox, the above information can be obtained by a function named "stereo calibration" when all parameters of both cameras are loaded. Also it provides us another function named "stereo triangulation", which can make use of the result about R' and T' to get the relationship between the coordinates of Image Reference Frame (IRF) and CRF:

$$\begin{bmatrix} X_{c\_L} & X_{c\_R} \end{bmatrix} = stereo\_triangulation \left(x_L, x_R, om, T', M_{1\_L}, M_{1\_R} \right) \tag{10}$$

where, x_L and x_R are 2D pixel coordinates of the point which are going to be reconstructed in left and right images respectively; om is the character vector of the rotation matrix R', it satisfies the relationship $R' = rodrigues\,(om)$; T' is the translation vector; M_1 is the intrinsic parameters matrix of the cameras; and the output X_{c_L} and X_{c_R} are the 3D coordinates described in left and right CRFs.

This means that any 2D image point we pick up from pictures can be matched to a 3D point described in CRF. In other words, when a group image coordinates is input, a corresponding point in CRF can be guaranteed as output. Then a point in WRF can be calculated as following:

$$X_w = R^{-1} \cdot \left(X_c - T \right) \tag{11}$$

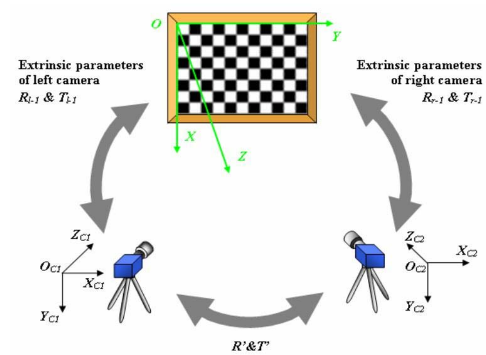


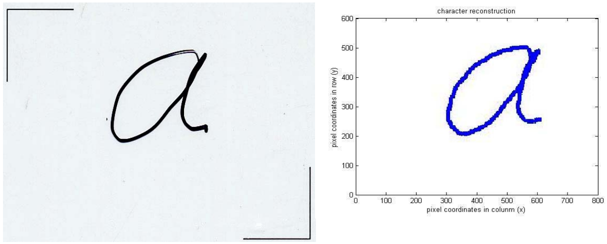
Figure 7. Schematics of multiple camera calibration experimental setup.

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.

IV. EXPERIMENTAL RESULTS

A. 2D Matching

Using the technique described in Section 2, we obtained the positions of the pen tip while it is moving on a transparent table during a hand-writing motion. According to these position coordinates obtained after matching, we first reconstructed an "*a*" character and compared it with the original character we wrote. As shown in Fig.8, it is clear that the two characters are similar enough to prove the position information would be reliable to carry out the calculation of velocity and acceleration.



Written character Reconstructed character

Figure 8. Comparison between original character and the reconstructed one.

In the above experiment, the sampling frequency of the camera was set to 200Hz, and the reconstructed "*a*" was made up of 280 points. The time interval between two neighboring points was 0.005s, which depends on the sampling frequency, and the total time of this motion was 1.4s. Thus, from Eq.6 and Eq.7, we can calculate both velocities and accelerations of the pen tip directly. The calculated velocities and accelerations in both x and y directions are shown respectively in Fig.9 and Fig.10. In the future, we will match the output of the MEMS acceleration sensors to acceleration information calculated by the techniques described above.

B. 3D Reconstruction

Fig.11 shows a pair of pictures taken by the left and right cameras respectively with the method of the one shown in Fig.7, and are selected to realize a 3D reconstruction. In this procedure, we took pictures of the template by the 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 on the template plane. The coordinates of two top points of the pen are calculated after stereo calibration, which were described in the world reference frame. The frame was defined by us as shown in Fig.11, 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.

In Fig.12 and Tab.1, both information about attitude and position described in the WRF are shown. For attitude result, there was approximately 2 degrees offset from the physical measurement. Measurement and pixel point selection errors might lead this offset. Since the error of length comparison was only 1.6%, the result of position calibration was good and acceptable.

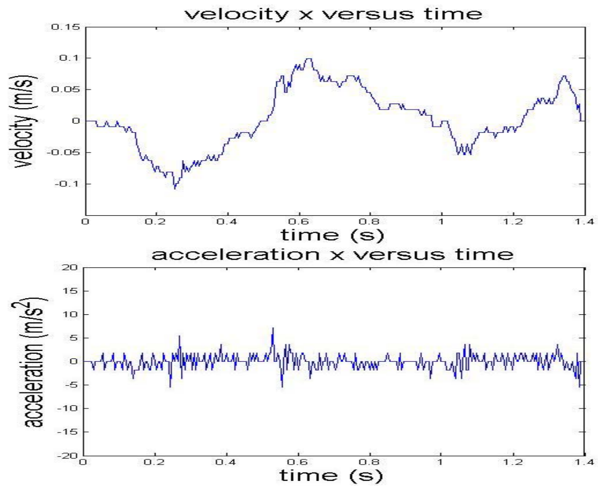


Figure 9. Velocity and acceleration in x direction.

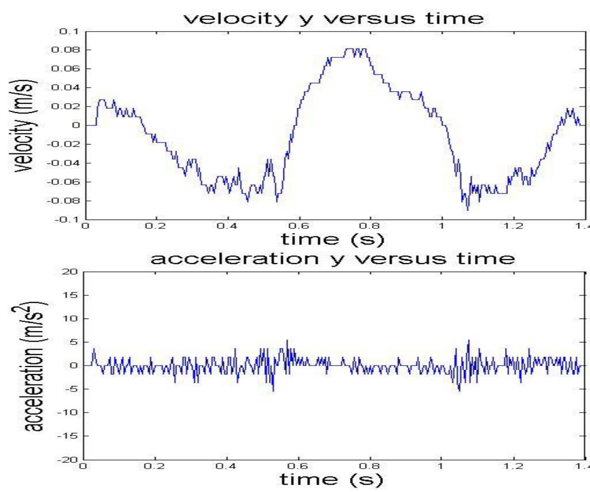


Figure 10. Velocity and acceleration in y direction.

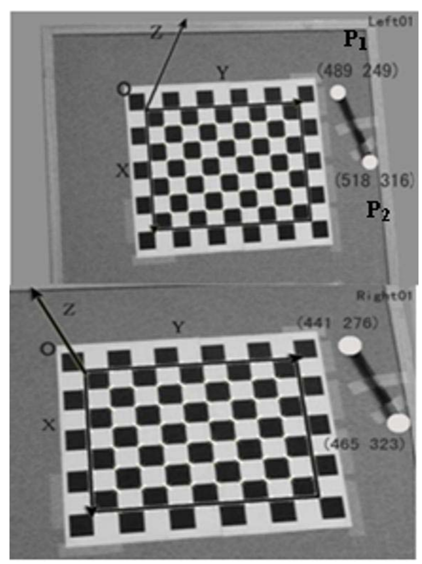


Figure 11. A pair of pictures for 3D reconstruction after camera calibration.

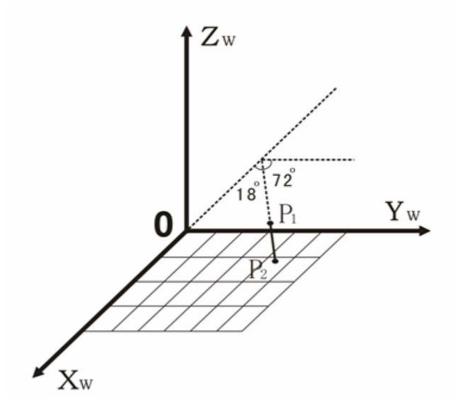


Figure 12. Experimental description by projection relationship.

	Physical Measurement Results	Calculated Results from 3D Reconstruction	
Angle between X axis (°)	18	16.58	
Angle between Y axis (°)	72	73.44	
Angle between Z axis (°)	90	89.23	
Length (*mm*)	140	137.76	
	X	Y	Z
Position of Point1 (*mm*)	-15.04	335.15	10.62
Position of Point2 (*mm*)	116.99	374.41	12.47

Table 1. Results of attitude and position based on multiple camera calibration with the parameters of the left camera.

V. CONCLUSION

This paper presents an entire design of calibration method for MEMS inertial sensors based on optical tracking techniques. We make use of block matching to estimate the motion vector of the pen tip. According to the change of the displacement, velocities and accelerations could be obtained using a high speed camera. Besides matching, based on camera calibration, multiple camera calibration is used to reconstruct a 3D point, and furthermore, calibrations of position, attitude and angular velocity can be carried out. A high speed camera plays an important role in this system. Our current experimental results illustrate that both block matching algorithm and multiple camera calibration are feasible to improve calibration reference for MEMS sensors.

However, the current attitude calibration part only includes the results on position and attitude. For future work, besides improving the current matching algorithms, we will focus on realizing all the functions, including the calibration of angular velocity.

ACKNOWLEDGMENT

We owe gratitude to the Hong Kong Innovation and Technology Commission (ITF-UIM-151) and DAKA Development Ltd. for sponsoring this project.

REFERENCES

[1] Guanglie 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Christian Wengert, “Camera Calibration Toolbox for Matlab”. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

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

[10] 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.

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

[12] 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.

[13] 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.

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