2D Human Gesture Tracking and Recognition by the Fusion of MEMS Inertial and Vision Sensors

Shengli Zhou, Fei Fei, Guanglie Zhang, John D. Mai, Yunhui Liu, Jay Y. J. Liou, and Wen J. Li, *Fellow, IEEE*

***Abstract*— In this paper, we present an algorithm for hand gesture tracking and recognition based on the integration of a custom-built microelectromechanical systems (MEMS)-based inertial sensor (or measurement unit) and a low resolution imaging (i.e., vision) sensor. We discuss the 2-D gesture recognition and tracking results here, but the algorithm can be extended to 3-D motion tracking and gesture recognition in the future. Essentially, this paper shows that inertial data sampled at 100 Hz and vision data at 5 frames/s could be fused by an extended Kalman filter, and used for accurate human hand gesture recognition and tracking. Since an inertial sensor is better at tracking rapid movements, while a vision sensor is more stable and accurate for tracking slow movements, a novel adaptive algorithm has been developed to adjust measurement noise covariance according to the measured accelerations and the angular rotation rates. The experimental results verify that the proposed method is capable of reducing the velocity error and position drift in an MEMS-based inertial sensor when aided by the vision sensor. Compensating for the time delay due to the visual data processing cycles, a moving average filter is applied to remove the high frequency noise and propagate the inertial signals. The reconstructed trajectories of the first 10 Arabic numerals are further recognized using dynamic time warping with a direct cosine transform for feature extraction, resulting in an accuracy of 92.3% and individual numeral recognition within 100 ms.**

***Index Terms*— Sensor fusion, gesture recognition, MEMS-based motion tracking, sensor calibration.**

I. Introduction

HUMAN gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head or body with the intent to convey meaningful information or to communicate with the environment [1]. With the rapid development of computer technology, human-computer interaction has become an ubiquitous activity in our daily life [2]. More attention has been focused on translating these human gestures into computer-understandable language in the past few years. Many gesture tracking and recognition technologies have been proposed. In general, these current gesture tracking technologies derive pose estimates from electrical measurements received from mechanical, magnetic, acoustic, inertial, optical, radio or microwave sensors [3]–[5]. Each sensor has its advantages and limitations. For example, mechanical sensors provide accurate pose estimates and have a low latency, but their mobility is low and they usually occupy a large volume of space. Magnetic sensors are also accurate for pose estimation, have a low latency, and good mobility [6]. But the problem is that they are vulnerable to distortions from conductive objects in the environment, and the signal attenuates quickly with an increase in distance between the magnet and the sensor. Acoustic sensors are small in size, lightweight and have good mobility, but their accuracy is affected by background ambient noise and, atmospheric effects. They also require a fairly unobstructed line-of-sight between the emitters and the receivers. MEMS-based inertial sensors are lightweight, good for fast motion tracking, and can cover a large sensing range, but they lack long term stability due to the problem of severe zero drift. Optical sensors are very accurate and have no accumulated errors, but their ability to resolve fast movements is poor due to motion blur. They also suffer from line-of-sight limitations. For radio and microwave sensing, they can cover a large tracking range and are very mobile, but their precision is low [3]. When applied to gesture recognition, most of these technologies can be used alone with good results, as summarized in Table I. But when it comes to gesture tracking, due to the limitations in accuracy, latency, noise, and tracking range, none of them are capable of tracking the motion perfectly. Recently, researchers have applied the fusion of multiple sensors to overcome the shortcomings inherent with a single sensor, and numerous papers on sensor fusion have been published in the literature. For example, multiple object tracking has been realized by fusing acoustic sensor and visual sensor data [1]. The visual sensor helps to overcome the inherent limitation of the acoustic sensor for simultaneous multiple object tracking, while the acoustic sensor supports the estimation when the object is occluded. "Inertial measurement units" (IMU's) which combine several types of sensors such as an accelerometer, a gyroscope, and a magnetometer, are used to track position and orientation of objects and are standard systems in many motion labs, [14]–[18]. In [16], the change in position and orientation is estimated by inertial sensors,

Manuscript received March 18, 2013; accepted October 10, 2013. Date of publication November 1, 2013; date of current version February 21, 2014. This work was supported in part by the City University of Hong Kong under Project 9666011, in part by the Hong Kong Applied Science and Technology Research Institute under Project 9211037, and in part by the Science Technology and Innovation Committee of Shenzhen Municipality under Project JCYJ20120618140504947. The author W. J. Li is indebted to The Chinese University of Hong Kong for its continual support of S. L. Zhou's graduate work. The associate editor coordinating the review of this paper and approving it for publication was Prof. Ralph Etienne-Cummings.

S. Zhou and Y. Liu are with the Chinese Univeristy of Hong Kong, Hong Kong (e-mail: slzhou@mae.cuhk.edu.hk; yhliu@mae.cuhk.edu.hk).

F. Fei, G. Zhang, J. D. Mai, and W. J. Li are with the City University of Hong Kong, Hong Kong (e-mail: feifei@cityu.edu.hk; gl.zhang@cityu.edu.hk; johnmai@cityu.edu.hk; wenjli@cityu.edu.hk).

J. Y. J. Liou is with the Hong Kong Applied Science and Technology Research Institute, Hong Kong (e-mail: jayliou@astri.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2013.2288094

TABLE I
SUMMARY OF GESTURE RECOGNITION WITH DIFFERENT SENSING TECHNOLOGIES IN RECENT YEARS

Technology	Recognition algorithm	Summary
Optical sensing (monocular camera) [8]	Fuzzy-c-mean clustering	Recognizes the 26 American Sign Language (ASL) alphabet from hand images. An average 93.23% recognition rate was achieved.
Optical sensing (stereo camera) [9]	Hidden Markov Model (HMM)	26 ASL alphabet and 10 Arabic numerals were recognized with a 98.33% recognition rate when 720 video samples were used for training and 360 video sequences for testing.
Inertial sensing [2]	Probabilistic neural network (PNN)	2-D handwritten digits were recognized with a recognition rate 98%, and 3-D hand gestures were recognized with a 98% rate.
Acoustic sensing [10]	Template matching	Recognized characters by identifying sound patterns generated from a writing instrument. Recognition rate of over 70% (alphabet) and 90% (26 words) were achieved.
Radio frequency sensing [11]	Multiple hypothesis approach	The motion patterns of passive RFID tags and hand gestures were tracked by using multiple hypothesis tracking and subtag count information with an accuracy of 93%.
Magnetic sensing [12]	Braille to ASCII mapping	A magnetic data glove was implemented for dynamic gesture recognition and motion tracking through interpretation of Braille-encode ASCII characters.
Inertial sensing and magnetic sensing [13]	Dynamic Time Warping (DTW)	95.7% recognition accuracy was obtained for their subject-dependent case, and 94.6% was obtained for their subject-independent case.

but when the estimated uncertainty associated with the relative position and orientation exceeds a predefined threshold, a magnetic measurement update is performed. By working together, an average rms error of 0.033 m for the position and 3.6 degrees for the orientation are obtained. GPS and inertial sensor fusion, and GPS and vision sensor fusion [19] have also been a hot research topic in the past few years [20], [21]. Sometimes these integrated systems are also aided with other additional sensors like velocity meters, magnetometers, and barometers [22], [23] to enhance tracking accuracy.

In this paper, we present results from our custom-built system that integrates a MEMS-based inertial sensor (i.e., consists of accelerometers and gyroscopes, and is sometimes refer to as an "μIMU") and a web-based camera (i.e., a vision sensor) for gesture tracking and recognition. For convenience of reference, we will call this μ**I**MU+**C**amera system the "μIC system" from here on. Even though hybrid tracking based on inertial sensors and vision sensors [24], [25] have been investigated in the past few years, the combination of gesture tracking and recognition via the fusion of an MEMS-based inertial sensor with a vision sensor fusion has not been frequently discussed. This paper includes the following: 1) the development of an algorithm that fuses inertial sensor and vision sensor data for simultaneously gesture tracking and recognition; 2) implementation of a noise update model based on sensor measurements so that the overall system is capable of judging which sensor data to utilize in different situations, and to dynamically adjust parameters according to the measured accelerations and angular velocities; and 3) an efficient feature extraction and gesture recognition algorithm, i.e., the proposed algorithm is capable of extracting the most important features from the trajectory and to project them from a higher dimension to a lower dimension with a high precision.

The rest of the paper is organized as follows. In Section II, we survey related work on the fusion of inertial sensor and vision sensor data for gesture tracking and recognition, and introduce the overall sensing system setup. This includes details about calibration of the sensors, the sensor fusion algorithm, the parameters used to tune the extended Kalman filter (EKF), and the system flow chart. In Section III, the experimental results with the proposed algorithm are discussed. In Section IV, the methodology for real-time gesture tracking is presented, and gesture tracking and recognition results are discussed. Finally, conclusions are presented in the last section.

II. VISION AND INERTIAL TRACKING

A. Related Work

For gesture *recognition*, high recognition rate can be obtained by independently using inertial sensors [5]–[26] or vision sensors [27], especially when real-time recognition is not required. But for real-time gesture *tracking*, inertial sensors suffer from the zero-drift problem while vision sensors have poor performance for resolving fast motions due to motion blur and occlusions. Hence, neither of them is perfect for gesture tracking alone. Hybrid gesture tracking base on vision and inertial sensor fusion offers not only fast motion tracking and good stability, but also robust performance over occlusions [28]. Gesture *tracking* has a wide range of real-world applications, such as augmented reality (AR) [29], surgical navigation [30], ego-motion estimation for robot or machine control in industry, and in helmet-tracking systems. Ego-motion estimation using a monocular camera, sampling at approximately 25 Hz, and an inertial sensor, sampling at 100 Hz, has been addressed in the literature [31], [32]. As reported in those work, an artificial planar object with seven known features was chosen for camera pose estimation.

The tracked 2D features from at least two different images were used to obtain the 3D position of the feature by linear triangulation. Measurements of inertial system were fused with measurements from the vision system by using a multi-rate Kalman filter without synchronization. With predefined process noise and measurement noise, the system demonstrated the ability to estimate the ego-motion of a sensor rig by fusing vision and inertial data. Furthermore, a hybrid EKF estimator that integrates a sliding window EKF and EKF-based SLAM, and an adaptive image-processing module that adjusts for the number of detected images were utilized for visual-aided inertial navigation as reported in [33]. These reported experimental results indicate that the proposed estimation framework in the next section is capable of real-time processing of image and inertial data on a typical microprocessor found in current mobile phones, in real-time. In [34], two web cameras, three gyroscopes and three accelerometers were used for the tracking and control of a quadrotor helicopter. Four active markers were precisely designed to improve visibility and robustness towards disturbances in their image-based pose estimation. Moreover, position and heading controllers for the quadrotor helicopter were implemented to show the system's capabilities, and the performance of the controllers was further improved by the use of onboard inertial sensors.

Existing hybrid methods differ in the number of inertial sensors and vision sensors, the performance specifications of the sensors, the number of visual feature points, the expression of the measurements, as well as the fusion algorithms. For a hybrid positioning system, the Kalman filter [35] which is the most widely used technique for implementing a Bayesian filter, and is generally used to estimate the system's position from sensor noise [36]. Since the application of a Kalman filter to a nonlinear system can be difficult, the extended Kalman filter (EKF), the unscented Kalman filter (UKF) [37] and the particle filter (PF) have been developed to deal with nonlinear problems. A comparison of the EKF, UKF and PF has been made by estimating movement using real data in [38]. Experiments show that these three methods yield similar estimation results, but the computational cost of the PF is much more than for the EKF and UKF. Furthermore, the computational cost of the UKF is about seven times higher than the EKF [32]. Thus, the EKF will give acceptable results with the least computational cost. Besides optimizing for an efficient algorithm, it is always desirable to use fewer, cheaper, smaller sized and simpler sensors to achieve the same goal. Therefore, we have built our motion tracking system by using a single low resolution camera commonly used for internet communications (i.e., a webcam) and a MEMS-based inertial sensor to capture the motion. The sampling frequency of the webcam is only 5 fps, which decreases the computational load on the system, but still gives good experimental results when used with an efficient algorithm for hand gesture tracking or recognition.

In order to address the synchronization problem due to the different sensor data acquisition frequencies, we have developed an algorithm for calculating the optimum filter length for a moving average filter. It not only helps to remove the high frequency noise, but also propagates the inertial data to solve the data synchronization problem.

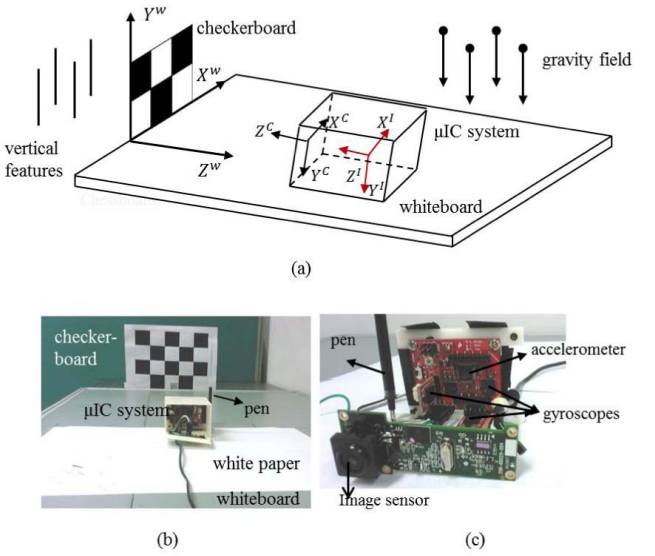


Fig. 1. In (a), the world coordinate frame is expressed with a superscript W. The camera coordinate frame is expressed with a superscript C, and the inertial coordinate frame is expressed with a superscript I. In (b), the experimental setup contains a 4 × 3 checkerboard, a μIMU and camera (μIMU) system, a pen with a piece of white paper for trajectory recording, and a whiteboard. The μIMU system in (c) contains a three axis accelerometer, three single axis gyroscopes, and an imager.

B. Experimental Setup

The experimental setup, as shown in Fig. 1, consists of one 4 × 3 checkerboard pattern, one CMOS image sensor (Logitech QuickCam Pro 9000), a three axis MEMS accelerometer (Freescale MMA7260 accelerometer) and three MEMS single-axis gyroscopes (LISY300AL gyroscope). The sampling rate of the μIMU is 100 Hz. The maximum frame rate of the imager is 30 fps but is reduced to 5 fps for this study. The imager and the μIMU are fixed inside a box, so their relative position will not be changed during the experiments. A pen is attached to the outside of the box so that the trajectory of the box will be recorded during the movement. Then the trajectory of the camera will be recovered from the recorded trajectory of the pen.

There are two main approaches for visual trajectory tracking: one is recognition-based, and the other is motion-based. We choose recognition-based visual tracking because the accumulated error is bounded in this situation. Even though motion-based approaches, which detect motion through optical flow tracking and motion-energy estimation are easier to use, they cannot be used if the camera motion is more than a few pixels [39]. Moreover, they are subject to noise, leading to imprecise values and the pixel motion is often detected but not quantified [39]. The detailed dimensions of the μIMU and camera (μIC) system, the grid size of the checkerboard, and the dimensions of the whiteboard are recorded in Table II.

TABLE II
DIMENSIONS OF THE EXPERIMENTAL SETUP

Devices	μIMU + camera (μIC)	Grid of chessboard	Whiteboard
Dimensions	80 mm×70 mm×53 mm	47 mm×44 mm	1000 mm×700 mm

C. Inertial Sensor and Vision Sensor Calibration

The drift rate depends largely on the minimization of the μIMU residual errors. If these errors are minimized, then the drift problem will be greatly reduced. Among all the sources of error, the constant bias and calibration error (including the scale factor and alignment) are the dominant error components. The constant bias for an accelerometer is the offset of its output signal from the true value. It is often estimated by measuring the long term average of the accelerometer's output when it is not undergoing any acceleration [40]. A six-position static and rate test calibration method is utilized to estimate the constant bias and scale factor [41]. This requires that the inertial system be mounted on a leveled surface with each sensitive axis of every sensor be pointed up and down in an alternating manner. However, in practical situations, perfect alignment is usually not possible without the aid of some reference devices.

In this paper, we proposed an easy and convenient inertial sensor calibration method to estimate the constant bias and the scale factor for the inertial sensor. The basic principle is that the norm of the static accelerometer outputs should equal gravitational acceleration no matter which direction the sensor is pointed. The preliminary constant bias b in each axis is estimated using the following equations:

$$b = (x_{up} + x_{down})/2 \tag{1}$$

where x_{up} and x_{down} are the average sensor measurements when the axis is pointed upward and downwards, respectively. The constant bias and scale factors are further refined by finding the minimum error ε between the norm of the sensor measurements and gravitational acceleration g, as expressed in the following equation

$$\sum_{i=1}^{3} (s_i(x_i - b_i))^2 - g^2 \leq \varepsilon \tag{2}$$

where s_i is the scale factor in each axis. The accelerometer is calibrated by taking measurements at six different static states. Finally, the averaged value is calculated.

For the relative pose calibration between the inertial and the visual sensors, we use the method mentioned by [42]. Since the relative translation can be measured directly, we only focus on the relative rotation. The relative rotation between the inertial frame and camera frame is calibrated by measuring the vertical direction using both sensors. The relationship between the gravity field, vertical features and the μIC system is illustrated in Fig. 1(a). This problem will become one of finding the rotation quaternion such that

$$\max(\mathbf{q}^T (\sum_{i=1}^{n} {}^{I}\mathbf{V}_i^T {}^{C}\mathbf{V}_i)\mathbf{q}) \tag{3}$$

where ${}^{I}V$ and ${}^{C}V_i$ are the quaternion matrix where gravity is measured by the inertial sensor and the vertical vanishing point is measured by the camera.

By solving (3), the relative rotation between the μIMU and CMOS imager can be determined. Once this calibration is done, the result can be used indefinitely, as long as the relative pose has not been changed.

D. Motion and Measurement Model

The ultimate objective of our project is to estimate the 3D position and the orientation of a moving device in the world coordinate system. The inputs to this system are the 3D accelerations and the angular rates from the μIMU and pose estimation from the vision sensor. The general nonlinear state-space models for the EKF equation are described as follows:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x_{k-1}}, \mathbf{w}_{k-1}) \tag{4}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \tag{5}$$

where $\mathbf{x}_k$ denotes the system states at time k, $\mathbf{z}_k$ denotes the observation vector from inertial or vision measurements, $\mathbf{f}[\bullet]$ is the system transition function, $\mathbf{h}[\bullet]$ is the measurement function, $\mathbf{w}_k$ is the zero-mean Gaussian process noise, with $\mathbf{w}_k \sim N(0, \mathbf{Q})$, and $\mathbf{v}_k$ is the zero-mean Gaussian observation noise, with $\mathbf{v}_k \sim N(0, \mathbf{R})$.

In order to project the state and covariance estimates from the previous time step *k-1* to the current time step k, the following time update equations are applied [35]

$$\hat{\boldsymbol{x}}_{\mathbf{k}}^{-} = \mathbf{f}(\hat{\boldsymbol{x}}_{k-1}, 0) \tag{6}$$

$$\mathbf{P}_k^{-} = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{Q}_{k-1} \tag{7}$$

where $\mathbf{A}_k$ is the transition matrix from Jacobian linearization:

$$\mathbf{A}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\hat{\boldsymbol{x}}_{k-1}, 0) \tag{8}$$

The equations for the measurement update are

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{k}}^{-} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^{-} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{9}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^{-} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\boldsymbol{x}}_k^{-}) \tag{10}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^{-} \tag{11}$$

where $\mathbf{K}_k$ is the Kalman filter gain and $\mathbf{H}_k$ is the measurement matrix from Jacobian linearization

$$\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}_k, 0) \tag{12}$$

E. Kalman Parameters

The states for EKF are defined as follows:

$$\mathbf{x} = [\mathbf{p^T}\ \mathbf{v^T}\ \mathbf{a^T}\ \mathbf{q^T}\ \boldsymbol{\omega}^{\mathbf{T}}]^T \tag{13}$$

where $\mathbf{p}$ denotes the position vector, $\mathbf{v}$ denotes the velocity vector, $\mathbf{a}$ denotes the acceleration vector, $\mathbf{q}$ represents the orientation quaternion, and $\boldsymbol{\omega}$ represents the angular rate vector. The quaternion is chosen to represent the orientation instead of the rotation matrix and the Euler angles because the singularity problem can be avoided by using a quaternion.

The dynamic equations for the translational components of the state vectors are

$$p(k+1) = p(k) + v(k)\,\Delta t + \frac{1}{2}a(k)\Delta t^2 \tag{14}$$

$$v(k+1) = v(k) + a(k)\Delta t \tag{15}$$

$$a(k+1) = a(k) \tag{16}$$

The dynamic equation of a rotation quaternion has the form

$$\mathbf{q}(k+1) = \left(\cos\left(\frac{|\omega|\,\Delta t}{2}\right)\mathbf{I} + 2\frac{\sin\left(\frac{|\omega|\Delta t}{2}\right)}{|\omega|}\mathbf{F}_{\mathbf{q}}(\omega)\right)\mathbf{q}(k) \tag{17}$$

where

$$\mathbf{F}_q(\omega) = \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{18}$$

and $\omega(t) = (\omega_x, \omega_y, \omega_z)^T$ is the angular velocity.

Eq. (17) is derived from the expression of the time evolution for a quaternion

$$\frac{d\mathbf{q}}{dt} = \frac{1}{2}\mathbf{q} * \bar{\boldsymbol{\omega}} \tag{19}$$

where $\bar{\boldsymbol{\omega}}(t) = (0, \omega_x, \omega_y, \omega_z)^T$ is the augmented angular velocity vector.

When a visual measurement is available, the output vector consists of the position and orientation of the moving device with respect to the world coordinate frame, so the output vector can be expressed as follows:

$$\mathbf{z_C} = [p_x \quad p_y \quad p_z \quad q_0 \quad q_1 \quad q_2 \quad q_3]^T \tag{20}$$

The corresponding measurement matrix that projects the states to the visual measurements is

$$\mathbf{H_C} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{I}_{4\times3} & \mathbf{0}_{4\times3} \end{bmatrix}_{7\times16} \tag{21}$$

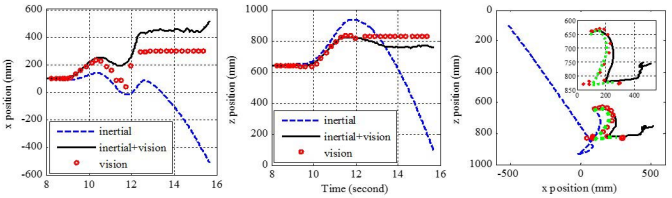
When an inertial measurement is available, the output vector will only contain 3D accelerations and angular rates, so the output vector can be expressed as follows:

$$\mathbf{z}_I = \begin{bmatrix} a_x & a_y & a_z & \omega_x & \omega_y & \omega_z & 0 \end{bmatrix}^T \tag{22}$$

The last zero is a dummy input to keep the output vector and the measurement matrix at the same size for both inertial measurements and vision measurements. The corresponding measurement matrix that projects the states to the inertial measurements is

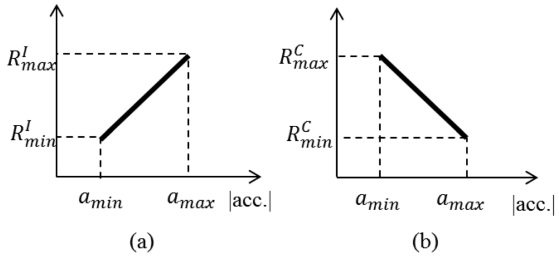
$$\mathbf{H_I} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \end{bmatrix}_{7\times16} \tag{23}$$

Using the algorithm proposed above, we conducted an experiment by writing the number "*2*" using the μIC system and applied the proposed algorithm to reconstruct the trajectory. The experimental results from using only the inertial sensor, both the inertial sensor and the vision sensor, and only the vision sensor are plotted in Fig. 2. The ground truth was obtained from the trajectory drawn by the pen attached to the device.



(a) x axis result. (b) z axis result. (c) Trajectories comparison.

Fig. 2. Comparison of trajectory tracking by using only the inertial data, both inertial data and vision data, and only the vision data. In (c), the green dotted line represents the reference trajectory.



(a) (b)

Fig. 3. "*Reliability*" update model for the inertial sensor and the vision sensor. (a) "*Reliability*" update of μIMU. (b) "*Reliability*" update of imager.

Since the relationship between the reference and time is unavailable, the reference is not plotted in Fig. 2(a) and Fig. 2(b). From Fig. 2(c) we can observe that the visual trajectory closely matches the reference, while the result only using the μIMU is the worst. Moreover, from Fig. 2(a) and Fig. 2(b) we find that even if a pre-calibration is done every time before the μIMU is switched on, the position from the μIMU continues to diverge from its real position. By fusing the inertial data with the vision data, this divergence has been greatly reduced.

F. Tuning the Covariance Matrix

A critical aspect of the EKF is the setting of the covariance matrices. As discussed before, the measurements of the performance of the inertial sensor improves with increased accelerations or angular velocities, which is contrary to the capabilities of the vision sensor. Based on this characteristic, we have developed an algorithm which is capable of determining the "*reliability*" of the inertial measurements and vision measurements through the calculation of measured accelerations and angular rates. When this "*reliability*" is applied to the Kalman filter, it is in the form of a measurement noise covariance. The measurement "*reliability*" update model for the μIMU and the imager is illustrated in Fig. 3. In Fig. 3(a) and Fig. 3(b), the accelerometer has the highest "*reliability*" and vision sensor has the lowest "*reliability*" when the absolute measured accelerations are at a maximum. But when the measured acceleration reaches the minimum, the accelerometer will have the lowest "*reliability*" while the vision sensor will have the highest "*reliability*".

Every time when the μIMU is powered on, the noise that drives the system will be different. So a calibration should be done to calculate the constant bias, the scale factor and the maximum noise covariance in the sensor.

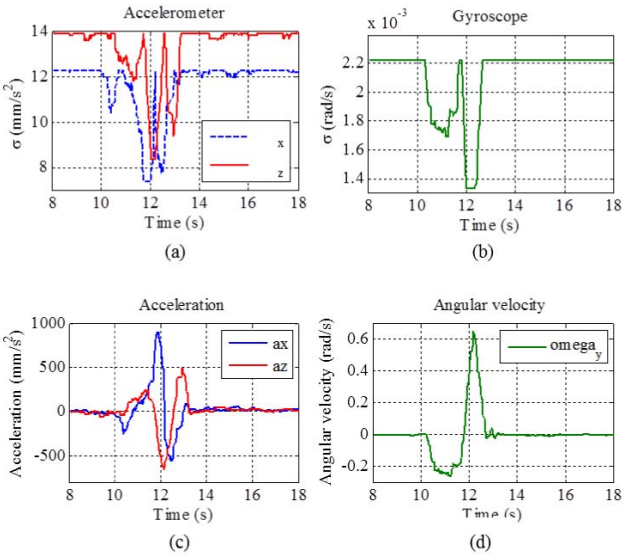


Fig. 4. Standard deviation in the measurement noise w.r.t. accelerations and angular rates. The upper two figures are the measurement noise standard deviation for the accelerometers and the gyroscopes, while the lower two figures are measured from accelerations and angular rates. (a) Standard deviation of measurement noise of accelerometer. (b) Standard deviation of measurement noise of gyroscope. (c) Accelerometer ouputs. (d) Gyroscope ouputs.

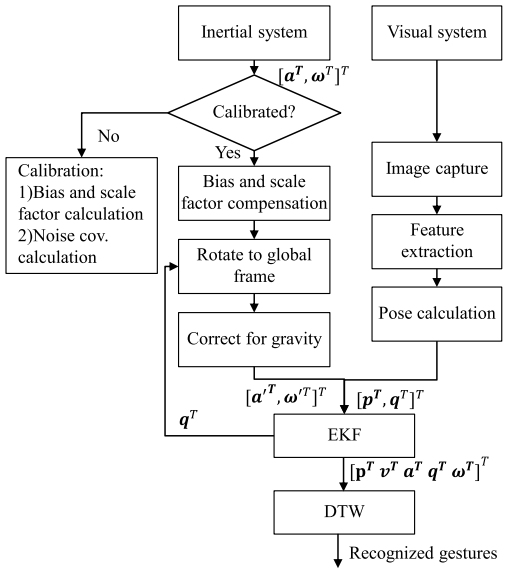


Fig. 5. Flow chart for the system with details for processing and fusing of data from the two sensors.

Suppose the measurement noise covariance of the acceleration at time t is $var\,(a, a) = \sigma_{at}^2$. From the relationship between velocity and time

$$v = a\Delta t \tag{24}$$

the noise covariance of the velocity is derived as

$$var\,(v, v) = \Delta t^2 \sigma_{at}^2 \tag{25}$$

From the time evolution of the position

$$p = v\Delta t + 0.5a\Delta t^2 \tag{26}$$

the noise covariance of the position can be computed as follows

$$\begin{aligned} var\,(p, p) &= \Delta t^2 var\,(v, v) + 0.25\Delta t^4 var\,(a, a) \\ &= 1.25\Delta t^4 \sigma_{at}^2 \end{aligned} \tag{27}$$

The same rules can be used for the gyroscope. Since the smallest measurement noise covariance for the IMU and the largest measurement noise covariance for the camera cannot be currently measured, these values are assigned from experimental experience. The standard deviation in the measurement noise with respect to measured accelerations and angular rates are shown in Fig. 4.

G. System Flow Chart

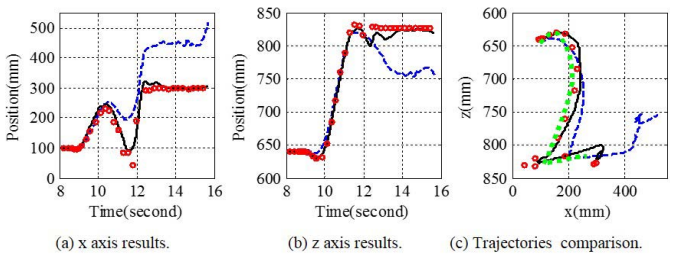
The inertial sensor and the vision sensor are integrated in a box as shown in Fig. 1. Once the system starts, the inertial frame is determined with respect to gravity, following the orientation obtained from the vision sensor. From the static inertial measurements and gravity in the inertial frame, the constant bias, the scale factor and the covariance in the measurement noise can all be computed. When this calibration is finished, the inertial data will first be compensated with the bias obtained during calibration, and then will be rotated to the global frame to correct for gravity. Finally, the inertial data will be sent to the EKF for pose estimation. For a vision system, an image of the checkerboard will first be captured, and then the corners of the image will be extracted. The pose of the camera w.r.t. the world coordinate system is then calculated using a direct linear transform (DLT); [43]. By fusing the inertial data and the vision data, the trajectory of the movement will be reconstructed. A dynamic time warping (DTW) is finally applied for the reconstructed trajectory recognition. The system flow chart is shown in Fig. 5.

III. Offline Algorithm Analysis

Offline analysis is necessary when the influences of the algorithm or parameters on the system need to be investigated. Thus, we can keep most parameters unchanged, and only investigate the ones of interested. In addition, since we know exactly when the inertial and visual data are taken, we can manually align the inputs to the filter. This solves the data synchronization problem in this case.

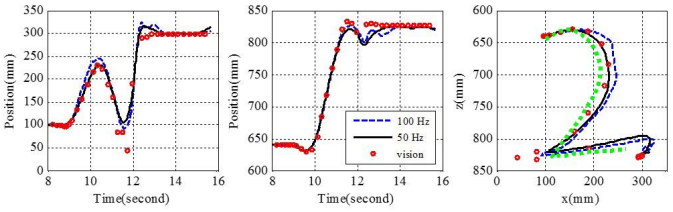
A. Experimental Results With Constant and Adaptive Parameter Updates

We have tested the proposed fusion algorithm by performing experiments with natural human movement, e.g. writing the Arabic numeral *2* using the experimental device. During this movement, both accelerations and angular velocities are needed for the trajectory tracking, so it is a more difficult problem than only pure translation or pure rotation. The ground truth is provided by the pen attached to the box. The experimental results with a constant parameter update and an adaptive parameter update are plotted in Fig. 6.



(a) x axis results. (b) z axis results. (c) Trajectories comparison.

Fig. 6. Experimental results with constant and adaptive parameter updates. The blue dashed line represents fusion result with a constant parameter update; the black solid line represents fusion result with an adaptive parameter update; the red circles represent positions from visual measurements; the green dotted line represent the reference trajectory.



(a) x axis results. (b) z axis results. (c) Trajectories comparison.

Fig. 7. Experimental results at 100 Hz and 50 Hz sampling frequencies. The blue dashed line represents fusion result when the inertial sensor was sampled at 100 Hz; the black solid line represents fusion result when inertial sensor was sampled at 50 Hz; the red circles represent positions from visual measurements; the green dotted line represents the reference trajectory.

From Fig. 6, we can see that the trajectory from the vision sensor best matches the ground truth for both the dynamic and static observations. When the actual movement is finished, the reconstructed trajectory with a constant parameter update is still increasing with time. Nevertheless, this situation has been greatly improved compared to the case where only the inertial sensor is used, but this is still problematic for practical applications. By comparing the results of the constant parameter update to the adaptive parameter update, we notice that the trajectory in the latter case is closer to the ground truth, and its static performance is better than the former case.

B. Experimental Results With Different Sampling Frequencies

The current sampling frequency of the inertial sensor is 100 Hz. The actual frequency of our hand motion is around 10 Hz, so the sampling frequency is much higher than needed. Moreover, more data output and processing means more power consumption. Therefore, the tracking results at a 50 Hz sampling frequency have also been examined. The experimental results when the sampling frequency of the inertial sensor is 100 Hz and 50 Hz are shown in Fig. 7.

From Fig. 7, we find that the system is capable of tracking the dynamic motion from about the 10^{th} second to the 12^{th} second, and is able to follow the visual measurements when the sensor stops moving. The 50 Hz sampling frequency reaches a steady state faster than the 100 Hz data. For the overall performance, the reconstructed trajectory at 50 Hz seems to be closer to the reference than the 100 Hz data.

IV. Real-Time Gesture Tracking Results

For real-time gesture tracking, the main difference from the offline analysis is the time delay. The time delay not only arises from the visual measurements but also comes from the inertial measurements. In this paper, we have proposed a simple method to deal with the time delay problem inherent to real-time gesture tracking.

A. Methodology to Handle Time Delay

A crucial part of sensor fusion is the data association, including the temporal synchronization of measurements from different sensors [44]. So it is critical to know exactly when the different measurements are taken. Since both of these sensors have their own clocks, they have to be synchronized. This can be achieved by initiating them simultaneously. However, clocks tend to diverge after a while, which will introduce more problems for long term operation [45]. In this paper, one reference clock signal is applied. Once the inertial measurement or visual measurement is available, the corresponding time from the reference clock will be recorded, and then the exact time interval between the two measurements is known.

However, there is always a delay between the time when the measurements are taken and the time that the measurement arrives at the filter, which is called the *measurement latency*. Several factors contribute to measurement latency, e.g. the measurement acquisition time, the pre-processing time, the communication transfer, and the data buffering [44]. These factors are not deterministic. Besides, due to the different data pre-processing schemes, the inertial measurement latency and visual measurement latency are different. So even if the exact time of the measurement is known, the data cannot be fused directly due to the time delay problem. In our experiments, the visual data always arrives at the filter 200 ms later than the inertial data, so the inertial data has to be propagated as it waits for the synchronous visual measurements.

We use a moving average filter to deal with this lag. A moving average filter is a simple and optimized method for reducing noise while keeping the sharpest step response [46]. The three general characteristics of all moving average filters are: (1) they smooth the input data, (2) they lag the original signal, and (3) they cut out the highest frequencies (i.e. they act as low pass filters) [47]. There are three popular types of moving average filters: a simple moving average (SMA), a weighted moving average (WMA), and an exponential moving average (EMA). We use a SMA due to its simplicity and computational efficiency. More importantly, most SMA algorithms introduce a time lag that roughly equals half of the length of the computational window. Suppose the current input to the Kalman filter is a pair of data $[t_n, a_n]$, where t_n is when the measurement is taken, and a_n is the acceleration at time t_n. The current visual input to the Kalman filter is also a pair of data $[t_v, p_v]$, where t_v is the time when the visual data is taken, and p_v is the measured position or orientation from the imager. Due to the time-consuming visual data processing, t_v is actually prior to t_n, that is, $t_v < t_n$. Next, suppose the filter length is L, then the lag will be $\frac{L-1}{2}dt$, where dt is the sampling period of the inertial sensor. The lag should be equal

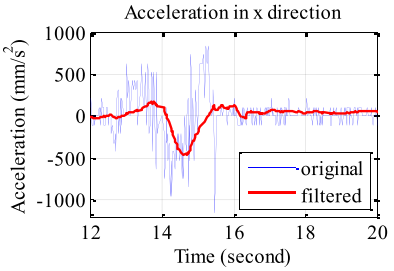


Fig. 8. Real-time acceleration output from the proposed algorithm.

to the current visual delay, that is,

$$\frac{L-1}{2}dt = t_n - t_v \tag{28}$$

so the filter length is

$$L = \frac{2 * (t_n - t_v)}{dt} + 1 \tag{29}$$

In our experiments, the time delay is about 200 ms, while the sampling period of the μIMU is 10 ms, thus the filter length should be around 41. This value is a roughly estimated value. Since the accelerations are not uniformly distributed, the lag will only be roughly located at the center of the filter window. The real-time acceleration output using the SMA is shown in Fig. 8.

B. Experimental Results for Trajectory Reconstruction

For the experiments, we wrote ten Arabic numerals and a cursive word with five English letters on a whiteboard using the μIC system. The corresponding experimental results are shown in Fig. 9 and Fig. 10. From Fig. 9, we notice that the reconstructed trajectories by using only inertial data are very different from the true trajectories. Especially for the static state after the movement is finished, the trajectories are still increasing with time. It is difficult to even recognize the trajectories from visual examination of the graphed data. For the results using only the vision sensor, due to the unstable performance of the imager, the estimated positions at some positions deviate too much from the real values, for example, for the number 2 and the number 3. Fortunately, by fusing the inertial and vision data, we can compensate for the individual disadvantages of the inertial sensor and the visual sensor. Even if the reconstructed trajectories are not exactly coincident with the ground truth, we can still easily recognize which numbers or characters correspond to which trajectories, from the black solid lines. The results are greatly improved compared to the results using only the inertial data. We have also reconstructed the more complex cursive trajectories of "*cityu*" in Fig. 10. The five characters are written in one continuous stroke.

The average velocities when writing these numbers are listed in Table III. The average velocity is calculated using the following expression:

$$\bar{v} = \frac{\sum_{i=m}^{n} \Delta s_i}{t_n - t_m} \bar{v} = \frac{\sum_{i=m}^{n} \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}}{t_n - t_m} \tag{30}$$

where m is the starting point of the number; n is the ending point of the number; t_i is the time at point i, Δs is the distance between two sampling positions. From Fig. 9 and Table III, we observe that the average velocities when writing number *0, 2, 3*, and *8* are greater than for the other numbers. In these cases, the vision sensor may lose the targets or incorrectly capture the pose due to the faster movement, i.e., numbers *2* and *3* in Fig. 9. The estimates at these points are corrected by using inertial data. For lower velocities, for example, writing numbers 1, 4, 5 and 7, noise dominates the inertial signals. Thus it is impossible to reconstruct the numbers by using inertial data only. But with the help of the relatively accurate visual data, the drift problem is greatly reduced. Thus, the overall system performance is improved.

C. Gesture Recognition With Dynamic Time Warping

In this paper, DTW is applied for trajectory-based gesture recognition. A detailed description of DTW can be found in [48]. The variety of handwriting style between different people makes character recognition a complex problem for a computer. In order to reduce the within-class variation, normalization has to be performed before implementing feature extraction and gesture recognition. Herein, we have implemented two normalization methods. Both of them are linear, but one is with the amplitude ratio preserved between two directions, and the other is with a fixed amplitude ratio.

1) Linear Normalization With the Amplitude Ratio Preserved (ARP): In this method, the trajectories in the x and z directions are projected to a 0.5×0.5 area through a linear transformation

$$x' = (x - x_{max})/Amp_{max} + Amp_x/2Amp_{max} \tag{31}$$

where Amp_x is the amplitude in the x direction; Amp_{max} is the largest amplitude in the x and z directions; x_{max} is the maximum value in x direction. The data in the direction with the bigger amplitude fills the range $[-0.5\ 0.5]$, while the data in the other direction are scaled according to their amplitude ratio. The normalization results for the ten Arabic numerals are shown in Fig. 11.

2) Linear Normalization With the Amplitude Ratio Fixed (ARF): The normalized trajectories using this method are also projected to a 0.5×0.5 area through a linear transformation. The difference is that the data fills both in the x direction and z direction rather than filling in only one direction.

$$x' = (x - x_{max})/Amp_x + 0.5 \tag{32}$$

The normalization results by using this method are plotted in Fig. 12.

For DTW, the templates for comparison are the real trajectories recorded by the pen during the experiments, the lengths of which are all 40. But for the test trajectories, the lengths depend on the writing speed. Even if DTW is capable of determining the optimum path between the templates and the testing trajectory, the redundant data will still influence the recognition accuracy and contributes to a heavy computational loading. Therefore, DCT is applied for feature extraction before normalization.

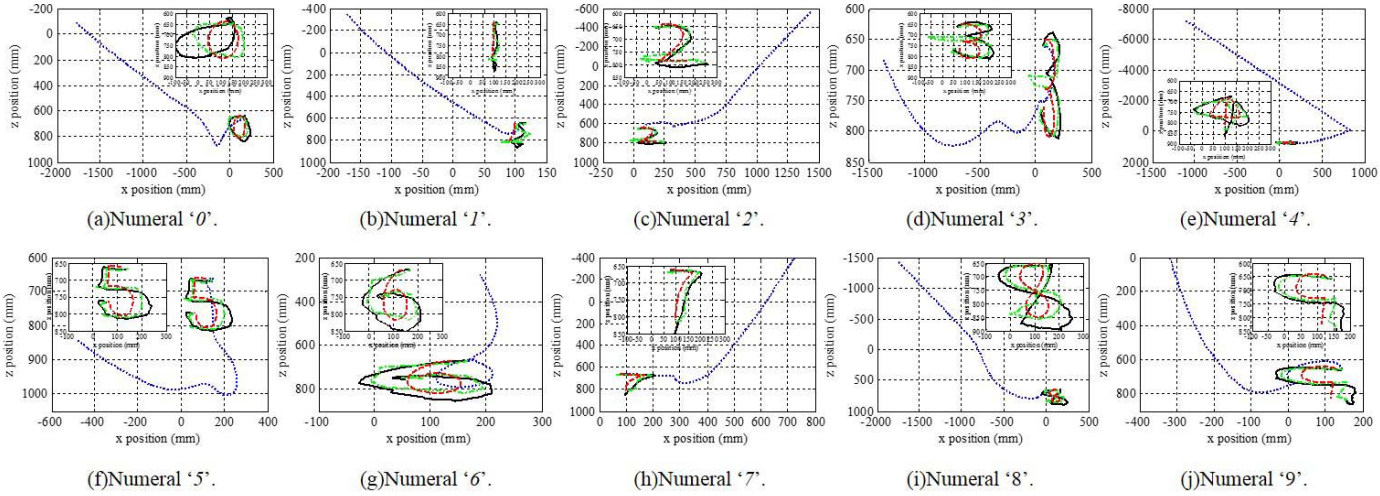


Fig. 9. Experimental results for number reconstruction. The blue dotted lines represent reconstructed trajectories by using the inertial sensor. The green circles represent positions from visual measurements. The black solid lines represent reconstructed trajectories by fusing inertial sensor and vision sensor data. The red dashed lines represent reference trajectories.

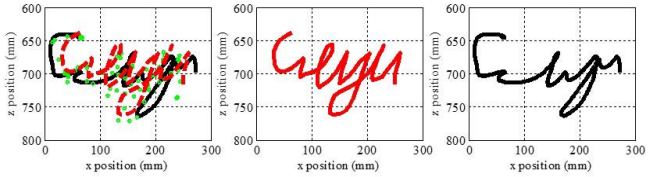


Fig. 10. Trajectory reconstruction of "*cityu*. In (a), the red dashed line represents the reference. The green circles represent positions from the vision sensor. The black solid line represents the fusion result.

TABLE III
AVERAGE WRITING VELOCITY OF THE ARABIC NUMERALS

No.	0	1	2	3	4	5	6	7	8	9
$\bar{v}$ (mm/s)	179.0	103.1	151.7	159.5	145.4	104.6	154.4	87.8	164.4	119.2

The DCT of a 1-D sequence of length N is

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad (33)$$

For $u = 0, 1, 2, \ldots, N-1$. In this frequency domain, the first 70 coefficients are kept, and then an inverse DCT transformation is performed to recover the trajectory. The inverse transformation is defined as [49]

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad (34)$$

For $x = 0, 1, 2, \ldots, N-1$. In both (33) and (34), $\alpha(u)$ is defined as

$$\alpha(u) = \begin{cases} \sqrt{1/N} & for \quad u = 0 \\ \sqrt{2/N} & for \quad u \neq 0 \end{cases} \quad (35)$$

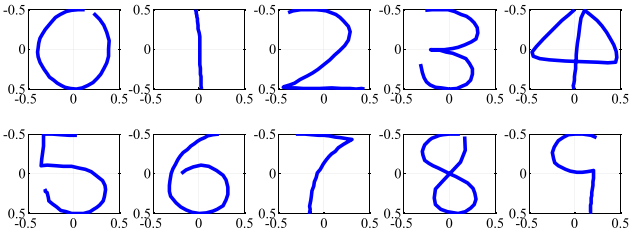


Fig. 11. Linear normalization with the amplitude ratio preserved.

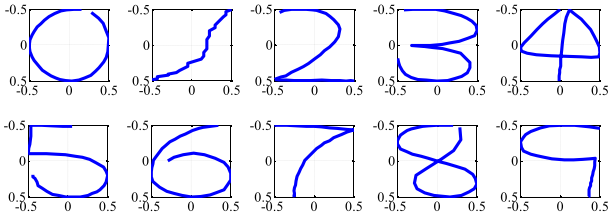


Fig. 12. Linear normalization with amplitude ratio fixed.

TABLE IV
EXPERIMENTAL RESULTS BY USING LINEAR NORMALIZATION AND LINEAR DCT

Algorithm	*ARP*	*ARF*	*ARP+DCT*	*ARF+DCT*
Accuracy (%)	76.9	73.08	92.30	92.30
Time Cost (ms)	438	443.7	104	102

The experimental results by applying two normalization methods and linear DCT are shown in Table IV.

From Table IV, it is found that a higher recognition accuracy is obtained by using ARP rather than by using ARF. When the linear DCT is applied, the accuracy of two normalization methods are both enhanced to 92.3%, and the computational cost is reduced by nearly 76%.

V. Conclusion

An algorithm has been developed to track the real-time position and orientation of a μIC system by fusing data from a MEMS-based inertial sensor and a vision sensor. The 100 Hz inertial data and the 5 fps vision data are fused by using an EKF. The measurement "*reliability*" is calculated based on the measured accelerations using a linear update model. Since tracking a motion that contains both translation and rotation is much more difficult than pure translation or pure rotation, we demonstrate that the algorithm is capable of reconstructing handwritten Arabic numerals and cursive words in real-time. The experimental results also prove that the reconstructed ten Arabic numerals can be recognized with an accuracy of 92.3% within 100 ms by using the DTW intuitive gesture recognition method.

Acknowledgment

Wen J. Li is also indebted to The Chinese University of Hong Kong for its continual support of S. L. Zhou's graduate work.

References

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst., Man, Cybern., Part C, Appl. Rev.*, vol. 37, no. 3, pp. 311–324, May 2007.

[2] J. S. Wang and F. C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2998–3007, Jul. 2012.

[3] D. H. Shin and W. S. Jang, "Utilization of ubiquitous computing for construction AR technology," *Autom. Construction*, vol. 18, no. 8, pp. 1063–1069, 2009.

[4] G. Welch and E. Foxlin, "Motion tracking: No silver bullet, but a respectable arsenal," *IEEE Comput. Graph. Appl.*, vol. 22, no. 6, pp. 24–38, Sep. 2002.

[5] S. Zhou, Q. Shan, F. Fei, W. J. Li, C. P. Kwong, P. C. K. Wu, *et al.*, "Gesture recognition for interactive controllers using MEMS motion sensors," in *Proc. 4th IEEE Int. Conf. Nano/Micro Eng. Molecular Syst.*, Jan. 2009, pp. 935–940.

[6] C. Hu, M. Li, S. Song, R. Zhang, and M. Q. H. Meng, "A cubic 3-axis magnetic sensor array for wirelessly tracking magnet position and orientation," *IEEE Sensors J.*, vol. 10, no. 5, pp. 903–913, May 2010.

[7] J. Lee, S. Hong, N. Moon, and S. J. Oh, "Acoustic sensor-based multiple object tracking with visual information association," *EURASIP J. Adv. Signal Process.*, vol. 2010, pp. 90–93, Aug. 2010.

[8] M. A. Amin and H. Yan, "Sign language finger alphabet recognition from gabor-PCA representation of hand gestures," in *Proc. 6th Int. Conf. Mach. Learn. Cybern.*, 2007, pp. 2218–2223.

[9] M. Elmezain, A. Al-Hamadi, S. S. Pathan, and B. Michaelis, "Spatio-temporal feature extraction-based hand gesture recognition for isolated american sign language and arabic numbers," in *Proc. 6th Int. Symp. Image Signal Process. Anal.*, 2009, pp. 254–259.

[10] A. Seniuk and D. Blostein, "Pen acoustic emissions for text and gesture recognition," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 872–876.

[11] P. Asadzadeh, L. Kulik, and E. Tanin, "Gesture recognition using RFID technology," *Pers. Ubiquitous Comput.*, vol. 16, no. 3, pp. 225–234, 2012.

[12] M. G. Ceruti, V. V. Dinh, N. X. Tran, H. Van Phan, L. T. Duffy, T. A. Ton, *et al.*, "Wireless communication glove apparatus for motion tracking, gesture recognition, data transmission, and reception in extreme environments," in *Proc. ACM Symp. Appl. Comput.*, 2009, pp. 172–176.

[13] C. Sung-do and L. Soo-Young, "3D stroke reconstruction and cursive script recognition with magnetometer-aided inertial measurement unit," *IEEE Trans. Consumer Electron.*, vol. 58, no. 2, pp. 661–669, May 2012.

[14] W. De Vries, H. Veeger, C. Baten, and F. Van Der Helm, "Magnetic distortion in motion labs, implications for validating inertial magnetic sensors," *Gait Posture*, vol. 29, no. 4, pp. 535–541, 2009.

[15] N. Phuong, H. J. Kang, Y. S. Suh, and Y. S. Ro, "A DCM based orientation estimation algorithm with an inertial measurement unit and a magnetic compass," *J. Univ. Comput. Sci.*, vol. 15, no. 4, pp. 859–876, 2009.

[16] H. M. Schepers, D. Roetenberg, and P. H. Veltink, "Ambulatory human motion tracking by fusion of inertial and magnetic sensing with adaptive actuation," *Med. Biol. Eng. Comput.*, vol. 48, no. 1, pp. 27–37, 2010.

[17] W. T. Faulkner, R. Alwood, D. W. A. Taylor, and J. Bohlin, "GPS-denied pedestrian tracking in indoor environments using an IMU and magnetic compass," in *Proc. Int. Tech. Meeting Inst. Navigat.*, 2010, pp. 198–204.

[18] A. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," in *Proc. IEEE Int. Symp. Intell. Signal Process.*, Aug. 2009, pp. 37–42.

[19] D. Dusha and L. Mejias, "Error analysis and attitude observability of a monocular GPS/visual odometry integrated navigation filter," *Int. J. Robot. Res.*, vol. 31, no. 6, pp. 714–737, 2012.

[20] J. Naranjo, F. Jiménez, F. Aparicio, and J. Zato, "GPS and inertial systems for high precision positioning on motorways," *J. Navigat.*, vol. 62, no. 2, pp. 351–363, 2009.

[21] C. H. Kang, S. Y. Kim, and C. G. Park, "Improvement of a low cost mems inertial-GPS integrated system using wavelet denoising techniques," *Int. J. Aeronautical Space Sci.*, vol. 12, no. 4, pp. 371–378, 2011.

[22] R. E. Hopkins, N. M. Barbour, D. E. Gustafson, and P. Sherman, *Miniature Inertial and Augmentation Sensors for Integrated Inertial/GPS Based Navigation Applications*. Cambridge, MA, USA, NATO RTO Lecture Series, 2010.

[23] J. Hol, "Sensor fusion and calibration of inertial sensors, vision, ultrawideband and GPS," Ph.D. dissertation, Linköping Studies Sci. Technol., Linköping Univ., Linköping, Sweden, 2011.

[24] G. Bleser and D. Stricker, "Advanced tracking through efficient image processing and visual-inertial sensor fusion," *Comput. Graph.*, vol. 33, no. 1, pp. 59–72, 2009.

[25] G. Bleser and G. Hendeby, "Using optical flow for filling the gaps in visual-inertial tracking," in *Proc. EUSIPCO*, 2010, pp. 1057–1063.

[26] R. Xu, S. Zhou, and W. J. Li, "MEMS accelerometer based nonspecific-user hand gesture recognition," *IEEE Sensors J.*, vol. 12, no. 5, pp. 1166–1173, May 2012.

[27] R. Z. Khan and N. A. Ibraheem, "Comparative study of hand gesture recognition system," in *Proc. Comput. Sci. Inf. Technol.*, 2012, pp. 203–213.

[28] Y. Tao, H. Hu, and H. Zhou, "Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation," *Int. J. Robot. Res.*, vol. 26, no. 6, pp. 607–624, 2007.

[29] Y. Suya, U. Neumann, and R. Azuma, "Hybrid inertial and vision tracking for augmented reality registration," in *Proc. Virtual Reality*, 1999, pp. 260–267.

[30] S. Giannarou, Z. Zhiqiang, and Y. Guang-Zhong, "Deformable structure from motion by fusing visual and inertial measurement data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2012, pp. 4816–4821.

[31] P. Gemeiner, P. Einramhof, and M. Vincze, "Simultaneous motion and structure estimation by fusion of inertial and vision data," *Int. J. Robot. Res.*, vol. 26, no. 6, pp. 591–605, 2007.

[32] L. Armesto, J. Tornero, and M. Vincze, "Fast ego-motion estimation with multi-rate fusion of inertial and vision," *Int. J. Robot. Res.*, vol. 26, no. 6, pp. 577–589, 2007.

[33] M. Li and A. I. Mourikis, "Vision-aided inertial navigation for resource-constrained systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2012, pp. 1057–1063.

[34] M. Achtelik, T. Zhang, K. Kuhnlenz, and M. Buss, "Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors," in *Proc. Int. Conf. Mechatron. Autom.*, 2009, pp. 2863–2869.

[35] G. Bishop and G. Welch, "An introduction to the Kalman filter," in *Proc. ACM SIGGRAPH Course 8*, Los Angeles, CA, USA, 2001.

[36] J. A. Corrales Ramón, F. A. Candelas Herías, and F. Torres Medina, "Kalman filtering for sensor fusion in a human tracking system," 2010.

[37] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Proc. AeroSense, 11th Int. Symp. Aerosp./Defense Sens., Simul. Controls*, 1997, pp. 182–193.

[38] L. Armesto, J. Tornero, and M. Vincze, "On multi-rate fusion for non-linear sampled-data systems: Application to a 6D tracking system," *Robot. Auto. Syst.*, vol. 56, no. 8, pp. 706–715, 2008.

[39] P. Saeedi, P. D. Lawrence, and D. G. Lowe, "Vision-based 3-D trajectory tracking for unknown environments," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 119–136, Feb. 2006.

[40] O. J. Woodman, "An introduction to inertial navigation," Dept. Comput. Lab., Univ. Cambridge, Cambridge, MA, USA, Tech. Rep. UCAMCL-TR-696, 2007.
[41] D. Titterton, J. Weston, D. Titterton, and J. Weston, *Strapdown Inertial Navigation Technology*. 2nd ed. London, U.K.: Inst. Electr. Eng., 2004.
[42] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *Int. J. Robot. Res.*, vol. 26, no. 6, pp. 561–575, 2007.
[43] G. T. Marzan and H. M. Karara, "A computer program for direct linear transformation solution of collinearity condition, and some applications of it," in *Proc. Symp. Close-Range Photogram. Syst.*, 1975, pp. 420–476.
[44] T. Huck, A. Westenberger, M. Fritzsche, T. Schwarz, and K. Dietmayer, "Precise timestamping and temporal synchronization in multi-sensor fusion," in *Proc. IEEE IV Symp.*, Sep. 2011, pp. 242–247.
[45] J. D. Hol, T. B. Schön, H. Luinge, P. J. Slycke, and F. Gustafsson, "Robust real-time tracking by fusing measurements from inertial and vision sensors," *J. Real-Time Image Process.*, vol. 2, no. 2, pp. 149–160, 2007.
[46] H. Sato, "Moving average filter," U.S. Patent no. 6 304 133, Oct. 16, 2001.
[47] J. F. Ehlers, *Rocket Science for Traders: Digital Signal Processing Applications*. New York, NY, USA: Wiley, 2001.
[48] P. Senin, *Dynamic Time Warping Algorithm Review*. New York, NY, USA: Wiley, 2008.
[49] S. A. Khayam, "The discrete Cosine transform (DCT): Theory and application," Dept. Electr. Comput. Eng., Michigan State Univ., Lansing, MI, USA, 2003.

Shengli Zhou received the B. Eng. degree in mechanical design, manufacture and its automation from Xidian University, China, in 2007, the M.Phil. and Ph.D. degrees from The Chinese University of Hong Kong (CUHK) in 2009 and 2013, respectively. From August, 2009 to July, 2010, she worked at the Hong Kong Applied Science and Technology Research Institute Company Limited (ASTRI). She is currently a Research Assistant with the Department of Electronic Engineering, City University of Hong Kong.

Fei Fei received the B.E. degree from the Department of Automation, University of Science and Technology of China (USTC) in 2004 and the M.E. degree from the Graduate School of the Chinese Academy of Sciences (CAS) in 2007. After receiving the Ph.D. degree from The Chinese University of Hong Kong (CUHK) in 2011, he has been working as a Postdoctoral Fellow in the Department of Mechanical and Biomedical Engineering, City University of Hong Kong.

Guanglie Zhang received the B. Eng. (1997) and M. Eng. (2000) degrees in electronic engineering, and the Ph.D. degree (2003) in control science and engineering, all from Xi'an Jiaotong University (XJTU), China. He has been a Visiting Assistant Professor with the Department of Mechanical and Biomedical Engineering, City University of Hong Kong, since May 2012. His current research interests are in the areas of micro/nano biotechnology, human-motion recognition algorithms, and complex-sensor-network fusion algorithms.

John D. Mai is currently a Visiting Assistant Professor with the Department of Mechanical and Biomedical Engineering, City University of Hong Kong, where his research focuses on optofluidics and micro-scale transport phenomena. He received the B.S. (1993), M.S. (1995), and Ph.D. (2000) degrees all in aerospace engineering from the University of California, Los Angeles, CA, USA. Afterwards, he was a Postdoctoral Fellow with the Department of Urology, Stanford University School of Medicine, USA, working towards the rapid detection of bacterial uropathogens. He is a Senior Member of the IEEE.

Yunhui Liu received the B.Eng. degree in applied dynamics from the Beijing Institute of Technology, Beijing, China, in 1985, the M.Eng. degree in mechanical engineering from Osaka University, Osaka, Japan, in 1989, and the Ph.D. degree in mathematical engineering and information physics from the University of Tokyo, Tokyo, Japan, in 1992. He worked with the Electrotechnical Laboratory, MITI, Japan, from 1992 to 1995. Since February 1995, he has been with the Chinese University of Hong Kong (CUHK) and is currently a Professor with the Department of Mechanical and Automation Engineering. He is a Visiting Professor at the State Key Lab of Robotics Technology and System, Harbin Institute of Technology, and the Director of Joint Centre for Intelligent Sensing and Systems of National University of Defense Technology and CUHK. He has published over 200 papers in refereed journals and refereed conference proceedings and is listed in the Highly Cited Authors (Engineering) by Thomson Reuters in 2013. His research interests include visual servoing, medical robotics, multi-fingered robot hands, mobile robots, sensor networks, and machine intelligence. Dr. Liu has received numerous research awards from international journals and international conferences in robotics and automation and government agencies. He is the Editor-in-Chief of Robotics and Biomimetics, a new journal published by Springer in 2014, and an Editor of Advanced Robotics, and served as an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and the general chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). He is Fellow of IEEE and HKIE.

Jay Y. J. Liou is Program Director of Applied Science and Technology Research Institute Company Limited (ASTRI) which is funded by the Hong Kong Government. Prior to ASTRI, He was the Senior R&D Director of UTStarCom (2001–2004) responsible for network modeling, traffic analysis and management. Between 1998 and 2001, he was R&D Director of Guoxin Lucent which is an joint venture between Bell Lab of Lucent Technologies and China Telecom. He also held various positions within Bell Lab AT&T and Lucent Technologies in USA. He received the Ph.D. degree from North Carolina, Chapel Hill, NC, USA, and the B.S. degree from National Taiwan University.

Wen J. Li (F'11) is a Chair Professor with the Department of Mechanical and Biomedical Engineering, City University of Hong Kong. He received the B.S.A.E. and M.S.A.E. degrees from the University of Southern California, Los Angeles, CA, USA, in 1987 and 1989, respectively, and the Ph.D. degree in aerospace engineering from the University of California, Los Angeles, in 1997. From September 1997 to October 2011, he was with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong. His industrial experience includes The Aerospace Corporation (El Segundo, CA, USA), NASA Jet Propulsion Laboratory (Pasadena, CA, USA), and Silicon Microstructures, Inc. (Fremont, CA, USA). His research interests include nanoscale fabrication, sensing, and manipulation. He currently serves as the Editor-in-Chief of the *IEEE Nanotechnology Magazine.*