Handwriting Trajectory Reconstruction Using Low-Cost IMU

Tse-Yu Pan, *Student Member, IEEE*, Chih-Hsuan Kuo, Hou-Tim Liu, and Min-Chun Hu, *Member, IEEE*

***Abstract*—In this paper, we propose a trajectory reconstruction method based on low-cost Inertial Measurement Unit (IMU) in smartphones. The IMU used in our work consists of a three-axis accelerometer and a three-axis gyroscope, which can record information of acceleration and rotation, respectively. Since intrinsic bias and random noise usually cause unreliable IMU signals, filtering methods are utilized to reduce high- or low-frequency noises of the signals. In addition, to more accurately detect whether the smartphone is moving or not, we extract multiple features from IMU signals and train a movement detection model based on linear discriminant analysis (LDA). Then, a "reset switch" mechanism is applied when the smartphone is detected as in a static state. The "reset switch" mechanism can effectively restrain the accumulated error of displacement calculation. Finally, the trajectory reconstruction results are applied to handwritten letter recognition for English alphabet, and the experimental results show that our proposed trajectory reconstruction method is reliable.**

***Index Terms*—Inertial measurement unit, complementary filter, linear discriminant analysis, trajectory reconstruction, handwritten letter recognition, inertial sensor, convolutional neural network.**

I. Introduction

THE rise of smart devices increases the popularity for end users to digitally record a variety of information by using the finger or the pen to type on a virtual keyboards or write in a specific area via touch screens. Compared to fingers, pen is more intuitive and precise to be used as a medium to make recordings such as signatures, notes, or paintings. There are still non-negligible differences between using a pen and using a finger as a medium to record trajectories [1]. Considering that a pen is a more precise and efficient tool to write complicated trajectories, devices such as styluses and graphics tablets have been developed to record trajectories. Unfortunately, the writing space is often too small for many handwriting applications. Thus recently, digital pens have been developed for the user to write on normal paper while the trajectories are simultaneously digitized and recorded. To digitally record these writing trajectories, specific sensors such as infrared sensors, ultrasonic sensors [2] or tiny cameras [3] are used on these digital pens. However, these trajectory recording devices usually require extra sensors or references in the using environment in addition to the sensors on the pen, which again limits the writing space. Furthermore, the above-mentioned sensors are high power-consumed sensors which cause a significant decrease in battery life. Therefore, some experts attempt to record movement data from IMU sensors without any other extra infrared or radar sensor in the environment.

In the past decades, IMU sensor has been widely used in the field of positioning [4], [5] and further utilized to achieve various HCI applications such as human activity recognition [6], gait detection [7], patient resuscitation [8], and sports sciences [9]. For example, Noitom Ltd.[1] developed a product named Perception Neuron, a set of IMU sensors worn on a human body, to record human movements and then control animation of virtual human in virtual space based on the recorded data. Thalmic Labs[2] introduced a wearable device named Myo, which is equipped with eight sEMG sensors and one IMU sensor to measure the trajectory of hand movements and simply recognize five basic hand gestures (i.e., fist, wave left, wave right, fingers spread, and double tap). However, due to noise problem of IMU, the above IMU-based researches or products only focus on large motion gestures/activities, which limits the development of HCI applications requiring subtle interactions in virtual reality (VR).

Therefore, in this work we propose a trajectory reconstruction method based on a low-cost IMU, which is usually equipped in smartphones. The IMU can directly sense the direction and the magnitude of movements, so it does not require extra sensors or references in the using environment. It also avoids the problem of limited writing space and the need for a writing medium. The main drawbacks of IMUs are the intrinsic drift and the high-degree noise, which make it difficult to reconstruct trajectories correctly. The goal of this paper is to solve this challenging problem and make a low-cost IMU in the smartphone can successfully reconstruct the handwritten trajectory when the user uses the smartphone as a digital pen. The proposed trajectory reconstruction method can be further applied to HCI applications involving subtle motion.

The contributions of this work are summarized as follows.

1) We design a framework which can not only reconstruct handwriting trajectory but also recognize the reconstructed letter (cf. Fig. 1) based on low-cost IMU. The

Manuscript received September 6, 2017; revised November 17, 2017 and December 21, 2017; accepted January 11, 2018. Date of publication March 8, 2018; date of current version May 23, 2019. This work was supported by the Ministry of Science and Technology, Taiwan under Contract MOST-103-2221-E-006-157-MY2 and Contract MOST 105-2221-E-006-066-MY3. *(Corresponding author: Min-Chun Hu.)*

The authors are with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: felix@mislab.csie.ncku.edu.tw; kuoe0@mislab.csie.ncku.edu.tw; timuncleuncle@mislab.csie.ncku.edu.tw; anita_hu@mail.ncku.edu.tw).

Digital Object Identifier 10.1109/TETCI.2018.2803777

[1] https://www.noitom.com/

[2] https://www.thalmic.com/

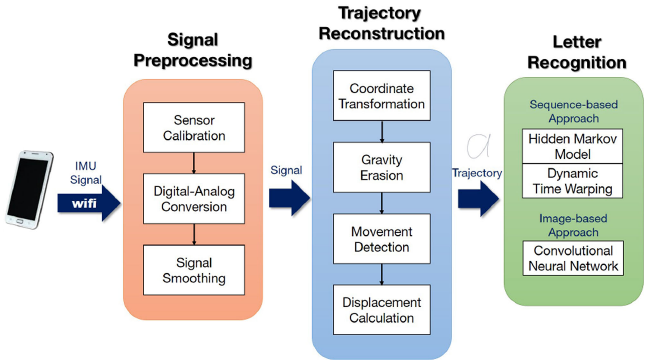


Fig. 1. Flowchart of the proposed trajectory reconstruction method and reconstructed letter recognition method.

proposed framework can be widely used in many IMU embedded devices and further applied to more HCI applications like gesture interaction in VR.

2) We propose a handwriting trajectory reconstruction method using low-cost IMU. To overcome the intrinsic drift and the high-degree noise, a reset switch method is designed to reduce the accumulated error caused by the IMU sensor. Moreover, LDA is applied to achieve more accurate movement detection results, which can consequently improve the reconstructed trajectory.

3) We investigate the performance of different letter recognition methods based on the reconstructed trajectories, including two sequence-based methods (i.e., Dynamic Time Warping (DTW) and Hidden Markov Model (HMM)) and an image-based approach (i.e., Convolutional Nerual Network (CNN)), and the CNN approach can achieve the best accuracy.

The remainder of this paper is organized as follows: Section II introduces the related work. Two parts of our method, i.e., the signal preprocessing part and the trajectory reconstruction part are expounded upon in Section III-A and Section III-B, respectively. Section IV shows the experimental results and conclusions are given in Section V.

II. Related Work

There have already been a number of products that were designed to reconstruct handwriting trajectory. For example, Wacom Inkling [2] used an external sensor to sense ultrasound and infrared to reconstruct the entire trajectory. Livescribe Echo [3] introduced a digital pen and a notebook to reconstruct the trajectory of handwriting by using invisible points on the notebook with which the pen could then locate itself by sensing the invisible points. Jin *et al.* [10] mounted an ultrasonic transmitter and an ultrasonic receiver on a pen device, and three reflectors were installed at the boundary of the writing space. The pen device transmited ultrasound and received the time sequence of the signals reflected by reflectors. The position of the pen device was then estimated by a trigonometry locating algorithm. Sperber *et al.* [11] proposed a vision-based digital pen to reconstruct the trajectory of handwriting. A tiny camera was mounted on the digital pen to capture the image of the paper area where the pen tip touched. There were some specific locating points produced by locality-sensitive hashing (LLAH), therefore the position of the digital pen can be determined through recognizing the pattern of these locating points in the captured image. However, above researches have to use extra sensors mounted in the environment to digitally record the trajectories.

With the popularity of inertial sensors, there have been many researches engaged in manipulating acceleration signals captured by accelerometer(s) [12]–[15] or dealing with both acceleration signals and angular velocities sensed by the IMU [16]–[19]. For example, Xie *et al.* [20] presented an accelerometer-based smart ring and used a similarity matching-based extensible algorithm to recognize basic and complex gestures. Xie *et al.* [21] demonstrated an accelerometer-based pen type sensing device. The Feedforward Neural Network (FNN) and Similarity Matching (SM) were utilized to recognize hand gestures based on the ACC signals. However, the trajectories of the hand gestures must be straight and with high divergence between each other. Wang *et al.* [22] proposed a handwriting recognition system to interpret the signals received by the accelerometer on a digital pen. They extracted certain features from those signals and selected the suitable ones by kernel-based classifications (KBCs). In their latest work, they proposed a real-time system to recognize the handwritten digits/letters in 3D space [23], [24] based on comparing the acceleration features of two signal sequences with the dynamic time warping (DTW) algorithm. Wang's method [23], [24] only recognized the handwriting but did not reconstruct the trajectory of handwriting. Agrawal *et al.* [25] proposed a method to reconstruct the trajectory of the handwriting by using the built-in accelerometer in a phone. However, with their method, the user has to write in the air with large movements to produce significant movement signals, which makes the system inconvenient to use.

It is quite challenging to reconstruct the trajectory accurately based on a single IMU because the intrinsic drift and the high-degree noise of IMU would significantly affect the reconstructed result. To deal with this problem, Yang *et al.* [26] proposed a mechanism named zero velocity compensation (ZVC) to reduce the accumulative error of IMUs when reconstructing the trajectory. Wang *et al.* [27] proposed an attitude error compensation method and a mechanism named Multi-Axis Dynamic (MAD) switch to discard some signals of noise, drift, and the users' trembles by setting an appropriate threshold. Sepahvand *et al.* [28] introduced a new Persian/Arabic handwritten character recognition method for an inertial-equipped sensor pen. The position signals were used to extract high-level features, and a metric learning technique was adopted to enhance the recognition accuracy. However, we observed that the above-mentioned trajectory reconstruction methods cannot work well on raw signals captured by a low-cost IMU sensor since the sensed signals involve much more noises. On the other hand, multiple features of IMU signals were extracted and fused to detect the movement of IMU in the above-mentioned works. The concept of feature fusion has been widely used in many detection/recognition applications. For example, Zhang *et al.* [29] proposed an ensemble manifold regularized sparse low-rank approximation (EMR-SLRA) algorithm for encoding multiple feature representations. Du *et al.* [30] presented a practical active learning algorithm to find a general way to choose the most suitable samples for training a classifier. We also take the advantages of feature fusion technique to detect the movement of IMU sensor, which is important for improving the trajectory reconstruction result.

In this work, we aim to develop a more accurate trajectory reconstruction method for a low-cost IMU, which is more widely equipped in mobile devices (e.g., smartphones). Moreover, we further investigate letter recognition methods to recognize the reconstructed trajectories. Two sequence-based methods (i.e., Dynamic Time Warping (DTW) and Hidden Markov Model (HMM)) and an image-based approach (i.e., Convolutional Nerual Network (CNN)) are compared in our work.

III. Methodology

In this section, we will introduce how to use signals measured by the IMU to reconstruct the sensor trajectory (cf. Fig. 1). The flow of the method can be divided into two parts, including the signal preprocessing part and the trajectory reconstruction part. In the signal preprocessing part, we apply sensor calibration, digital-analog conversion and smoothing filter on raw signals to reduce IMU noises. In the trajectory reconstruction part, attitude estimation, coordinate transformation, gravity erasion are applied on the preprocessed signals, and then movement detection and displacement calculation are utilized to obtain the final trajectory.

A. Signal Preprocessing

Given an IMU, the corresponding zero-g output (ZGO) and the sensitivity of each axis can be looked up according to the specification. Through these parameters, the digital signals can be converted to analog signals representing the physical acceleration. The zero-g output is the offset between zero and the output of a stationary IMU. The sensitivity is related to the full-scale range and the resolution of the IMUs. However, the IMU signals can easily be disturbed by a number of factors, such as the manufacturing process, internal thermal, mechanical fluctuations, and even the surrounding temperature. These factors result in the drift problem and white noises of the IMU signals, which make zero-g output and sensitivity different from the ones provided by the specification. Therefore, we apply a calibration mechanism to solve the drift problem and use a low-pass filter to deal with the white noises.

1) Sensor Calibration and Digital-Analog Conversion: The acceleration can be easily obtained by

$$a = \frac{Raw_{acc} - ZGO}{sensitivity_{acc}}, \tag{1}$$

where Raw_{acc} is the raw signal, ZGO is the zero-g output, and $sensitivity_{acc}$ is the sensitivity. Because of the gravity of Earth, the overall acceleration sensed by the sensor should be 1g, which will be dispersed into three axes. As shown in Fig. 2, the resultant vector of acceleration A should be on a sphere S centered on the origin $(0, 0, 0)$ with the radius equal to 1. However, since the zero-g output and the sensitivity might be affected by many factors mentioned in the previous paragraph, the vector A would not be $(0, 0, 0)$ when IMU is stationary. We apply the method proposed by Mark Pedley [31] to calibrate the accelerometer. It applies the concept of *ellipsoid to sphere mapping* to determine the real zero-g output and the sensitivity for each axis, and the calibrated signals are shown in Fig. 3. Similarly, the raw signals of gyroscope can be converted from digital to analog using the values of the zero-rate output (ZRO) and the sensitivity by

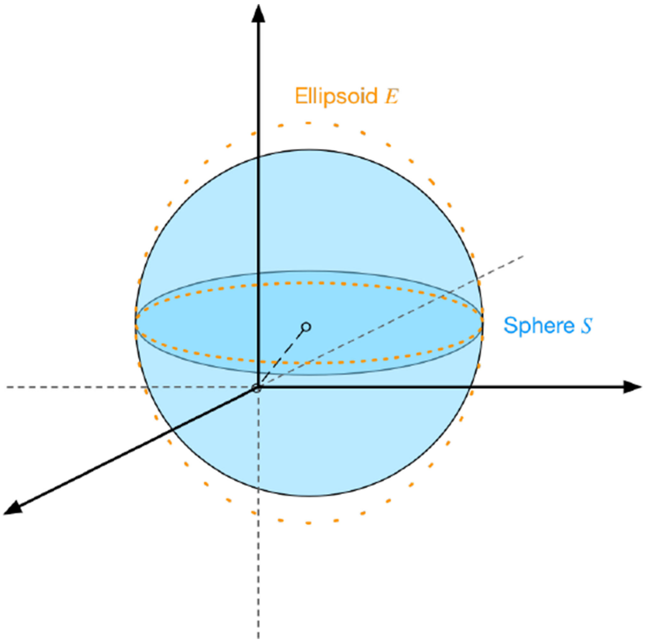


Fig. 2. Concept of ellipsoid to sphere mapping.

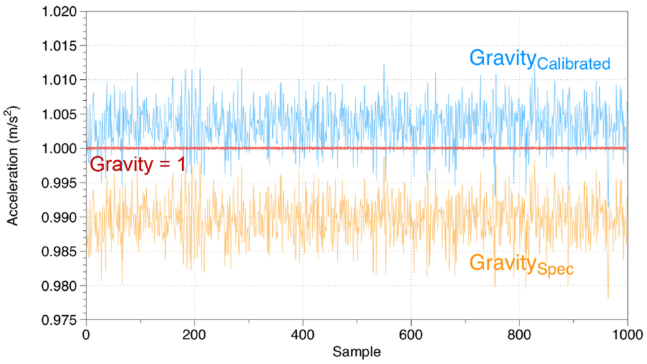


Fig. 3. The calibrated signals of the accelerometer after applying ellipsoid to sphere mapping.

$$\omega = \frac{Raw_{gyr} - ZRO}{sensitivity_{gyr}}. \tag{2}$$

However, unlike the accelerometer, the gyroscope is not affected by the gravity of Earth. Since there is no natural reference (i.e., the gravity of Earth) that can be utilized to calibrate the gyroscope, the value on the specification is directly used for the sensitivity of the gyroscope. The mean values of the initial signals are utilized to calculate a more accurate zero-rate output

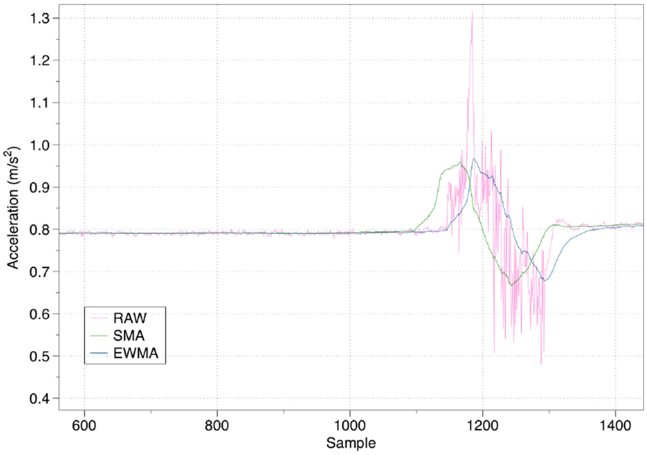


Fig. 4. The smoothing results of SWA and EWMA.

by

$$
\begin{aligned}
ZRO_X &= \bar{Raw}_X(k),\ k = 0...t_{init}\\
ZRO_Y &= \bar{Raw}_Y(k),\ k = 0...t_{init}\\
ZRO_Z &= \bar{Raw}_Z(k),\ k = 0...t_{init}
\end{aligned}
\tag{3}
$$

where t_{init} is a short duration in which the smartphone is static for collecting initialization signals.

2) Smoothing Filter: Since the IMU signals contain white noises which make the signals unreliable. Thus, we apply a low-pass filter to smooth the signals. Instead of using simple moving average (SMA) filter, we choose the exponential weighted moving average (EWMA) filter proposed by J. Stuart Hunter [32] because recent signals are more important than earlier signals. That is,

$$
EWMA(t) = \frac{Data_t + (1-\alpha)Data_{t-1} + (1-\alpha)^2 Data_{t-2} + \ldots}{1 + (1-\alpha) + (1-\alpha)^2 + \ldots}. \tag{4}
$$

where $Data_t$ is the raw signal of the IMU at time t and $\alpha = \frac{2}{N+1}$. N is set to be 50 in this work. Fig. 4 shows the result.

B. Trajectory Reconstruction

1) Attitude Estimation: When the user holds a smartphone, the coordinate system of the IMU will be different from the coordinate system of the real world, which makes the accelerations measured by the accelerometer inconsistent with the accelerations of the real movements. The accelerations would disperse into different axes according to the inclination of the IMU. As shown in Fig. 5, let X_w, Y_w, and Z_w denote the 3 axes of the real world coordinate system, and X_p, Y_p and Z_p denote the 3 axes of the IMU coordinate system, respectively. The calibrated accelerations of 3 axes are denoted as a_{X_p}, a_{Y_p}, and a_{Z_p}, and the calibrated angular velocities are denoted as ω_{X_p}, ω_{Y_p}, and ω_{Z_p}. The Euler angles ϕ, θ, and ψ represent the roll angle, the pitch angle, and the yaw angle, respectively. Based on these values, we can estimate the real attitude of the IMU.

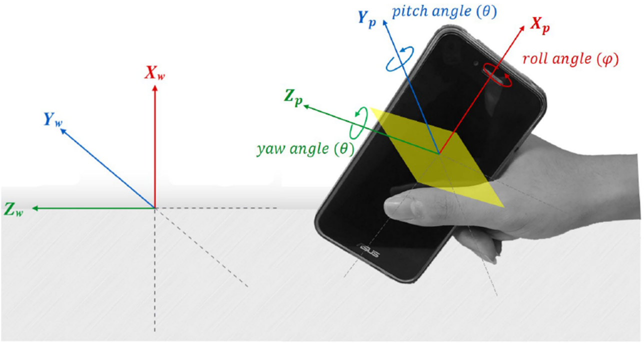


Fig. 5. The coordinate systems of the real world and the IMU.

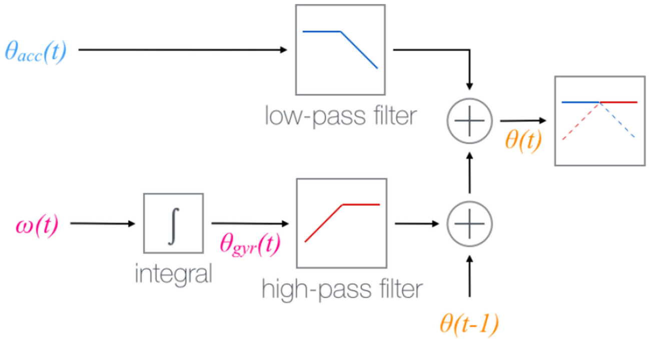


Fig. 6. The concept of complementary filter.

When the IMU is stationary, it is always affected by the gravity of Earth, which disperses its force into different axes according to the inclination angle between the IMU and the ground. The absolute attitude can be estimated by

$$
\begin{cases}
\theta_{acc} = \tan^{-1}\frac{a_{Z_p}}{a_{X_p}}\\
\psi_{acc} = \tan^{-1}\frac{a_{Y_p}}{\sqrt{a_{X_p}^2 + a_{Z_p}^2}}
\end{cases}. \tag{5}
$$

The gravity of Earth only helps us to infer the inclination angle between $Y_p Z_p$-plane and the $Y_w Z_w$-plane. There is no other information to infer the roll angle ϕ_{acc}, so we directly assume ϕ_{acc} as 0.

However, when the IMU is not stationary, the magnitude and direction of force it experienced will not be fixed. If we still use (5) to estimate the absolute attitude, the estimated result would change rapidly and be unreliable. In this case, we consider the angular velocity measured by the gyroscope to calculate the relative attitude. The attitude change ϕ_{gyr}, θ_{gyr}, and ψ_{gyr} can be easily obtained by integrating the measured angular velocity, that is,

$$
\begin{cases}
\phi_{gyr} = \omega_{X_p} \times dt\\
\theta_{gyr} = \omega_{Y_p} \times dt\\
\psi_{gyr} = \omega_{Z_p} \times dt
\end{cases} \tag{6}
$$

In order to reduce the error caused by the intrinsic property of IMUs, we apply the complementary filter, whose concept is shown in Fig. 6. The idea of complementary filter is to consider the attitude estimated by the accelerometer as signals full of high-frequency noises and the attitude estimated by the gyroscope as signals full of low-frequency noises. A low-pass filter is then applied to improve the attitude estimated by the

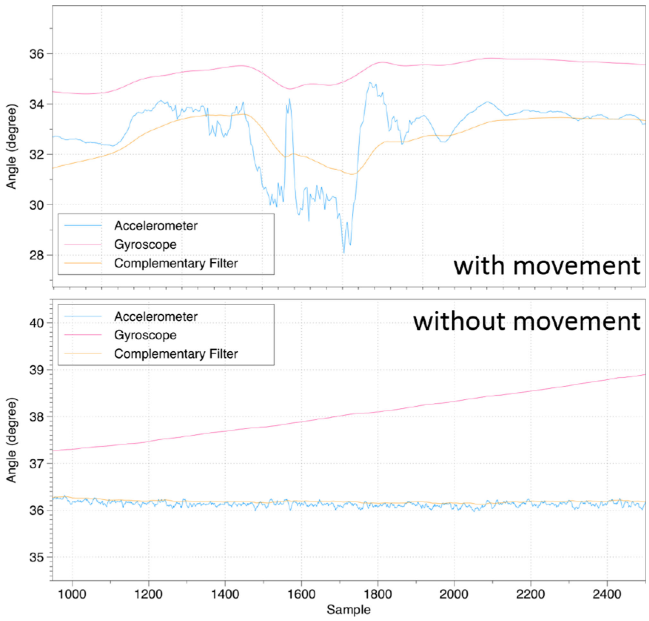


Fig. 7. An example of estimating attitude by accelerometer, gyroscope, and the signal using complementary filter, respectively. Upper: The estimated attitudes for an IMU with movement. Down: The estimated attitudes for an IMU without movement.

accelerometer and a high-pass filter is used for enhancing attitude estimated by the gyroscope. The final attitude is determined by combining the absolute attitude estimated by the accelerometer and the relative attitude estimated from the gyroscope. That is,

$$\theta(t) = (1-\alpha) \times (\theta(t-1) + \omega(t)\, dt) + \alpha \times \theta_{acc}(t) \quad (7)$$

where $\theta(t)$ is the attitude estimated by the complementary filter at time t, and $\theta(t-1)$ is the attitude estimated by the complementary filter at time $t-1$. $\theta_{acc}(t)$ is the absolute attitude estimated by the accelerometer, and $\omega(t)$ is the angular velocity measured by the gyroscope at time t. α is set to 0.2. Fig. 7 shows the estimated $\theta(t)$ by using the accelerometer, the gyroscope, and the complementary filter, respectively when the user holds the smartphone with or without movements. $\phi(t)$ and $\psi(t)$ can be estimated in a similar way.

2) Coordinate Transformation: Given the estimated attitude, the IMU coordinate system can be transformed into the world coordinate system by using quaternion representation, which is more accurate than directly rotating the 3 axes based on Euler angles. If we want to rotate a coordinate around a vector (x, y, z) with angle θ, the rotation quaternion would be $\cos(\frac{\theta}{2}) + x' \sin(\frac{\theta}{2})i + y' \sin(\frac{\theta}{2})j + z' \sin(\frac{\theta}{2})$, where vector (x', y', z') is the unit vector of the vector (x, y, z). The conversion between the Euler angle and the quaternion is formulated by

$$\begin{aligned} w &= \cos(\alpha)\cos(\beta)\cos(\gamma) + \sin(\alpha)\sin(\beta)\sin(\gamma) \\ x &= \cos(\alpha)\cos(\beta)\sin(\gamma) - \sin(\alpha)\sin(\beta)\cos(\gamma) \\ y &= \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\cos(\beta)\sin(\gamma) \\ z &= \sin(\alpha)\cos(\beta)\cos(\gamma) - \cos(\alpha)\sin(\beta)\sin(\gamma) \quad (8) \end{aligned}$$

where $\alpha = \frac{\psi}{2}$, $\beta = \frac{\theta}{2}$, and $\gamma = \frac{\phi}{2}$. The rotation formula based on quaternion is then defined by

$$P' = QPQ^{-1}, \quad (9)$$

where P' is the coordinate after rotation, and P is the original coordinate. Q is the rotation quaternion, and Q^{-1} is the inverse of quaternion Q.

3) Gravity Erasion: The IMU is always affected by the gravity of Earth. However, we do not need the acceleration caused by the gravity of Earth when reconstructing the trajectory. Theoretically, the gravitational acceleration should be 1g. Unfortunately, the acceleration signals measured by a stationary IMU are not always equal to 1g because of noises. Alternatively, we calculate the mean of the accelerations measured in an initial stationary period and take it as the real gravitational acceleration g_{init}. g_{init} is then subtracted from all accelerations by

$$\begin{bmatrix} a_{X_r} \\ a_{Y_r} \\ a_{Z_r} \end{bmatrix} = \begin{bmatrix} a_{X_p} \\ a_{Y_p} \\ a_{Z_p} \end{bmatrix} - \begin{bmatrix} g_{init} \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

4) Movement Detection: The movement detection step is important for the later displacement calculation step. We first segment the signals using a sliding window and extract features (as shown in Table I) for each segment. Instead of using a simple threshold to detect the smartphone movement as implemented in previous related works, these multiple features extracted from IMU signals are used for training a more robust movement detection model. Linear discriminant analysis (LDA) is utilized to classify a segment into moving or stationary signals, and morphological operations are then applied to smooth the classification results obtained by LDA. Fig. 8 presents an example of the classification results.

5) Displacement Calculation: The displacement is calculated and integrated to obtain the final trajectory by

$$\begin{aligned} displacement(t) &= displacement(t-1) \\ &\quad + velocity_{correct}(t)dt \quad (11) \end{aligned}$$

In the case of a perfect accelerometer, the velocity at the end of a stroke must be zero. However, in reality the velocity integrated from the estimated accelerations is not exactly zero because of noises that can not be eliminated in the preprocessing step. It consequently results in serious error in displacement calculation. To solve this problem, a reset switch mechanism based on the pre-mentioned movement detection method is used to compensate the velocity. If the smartphone is detected as moving, the switch will be turned off and the acceleration will be integrated directly to obtain the velocity. Otherwise, if the smartphone is detected as static, the switch is turned on and the velocity is reset to zero. However, as shown in Fig. 9, when the previous signal is detected as a moving signal, resetting the current velocity to be zero would make the velocity increase or decrease abruptly. The abrupt increase or decrease can be taken as the velocity error, which is caused by using the error-included acceleration estimations to accumulate the velocity value. In order to correct the velocity error and make the change of velocities continuous, we apply linear interpolation to the accumulated velocity. The

TABLE I
FEATURES FOR ALL SEGMENTS OF SIGNALS

Mean $\text{Mean} = \frac{1}{\lvert segment\rvert}\sum_{i=1}^{\lvert segment\rvert} a_i$	**Variance** $\text{VAR} = \frac{1}{\lvert segment\rvert}\sum_{i=1}^{\lvert segment\rvert} (a_i - mean(segment))^2$
Root Mean Square $\text{RMS} = \sqrt{\frac{1}{\lvert segment\rvert}\sum_{i=1}^{\lvert segment\rvert} a_i}$	**Standard Deviation** $\text{STD} = \sqrt{\frac{1}{\lvert segment\rvert}\sum_{i=1}^{\lvert segment\rvert} (a_i - mean(segment))^2}$
Maximum Absolute $\text{M}A_1 = \max([\lvert a_i\rvert, 1 \le i \le \lvert segment\rvert])$	**Minimum Absolute** $\text{M}A_2 = \min([\lvert a_i\rvert, 1 \le i \le \lvert segment\rvert])$
Maximum $\text{Max} = \max(\text{segment})$	**Minimum** $\text{Min} = \min(\text{segment})$
Interquartile Range $\text{IQR}{=}a_{q3} - a_{q1}$, where $q_k = \frac{\lvert segment\rvert}{4} \times k$	**Range** $\text{Range} = \max(\text{segment}) - \min(\text{segment})$
Median $\text{Median} = \begin{cases} a\frac{\lvert segment\rvert + 1}{2}, & \text{if } \lvert segment\rvert \text{ is } odd \\ \frac{a\frac{\lvert segment\rvert}{2} + a\frac{\lvert segment\rvert}{2} + 1}{2}, & \text{if } \lvert segment\rvert \text{ is } even \end{cases}$	**Mean Absolute Deviation** $MAD_1 = \frac{1}{\lvert segment\rvert}\sum_{i=1}^{\lvert segment\rvert} \lvert a_i - mean(segment)\rvert$
Max Mean Absolute Deviation $\text{M}MAD_1 = \max([\lvert a_i - mean(segment)\rvert, 1 \le i \le \lvert segment\rvert])$	**Max Median Absolute Deviation** $\text{M}MAD_2 = \max([\lvert a_i - mean(segment)\rvert, 1 \le i \le \lvert segment\rvert])$
Median Absolute Deviation $MAD_2 = \text{median}([\lvert a_i - mean(segment)\rvert, 1 \le i \le \lvert segment\rvert])$	

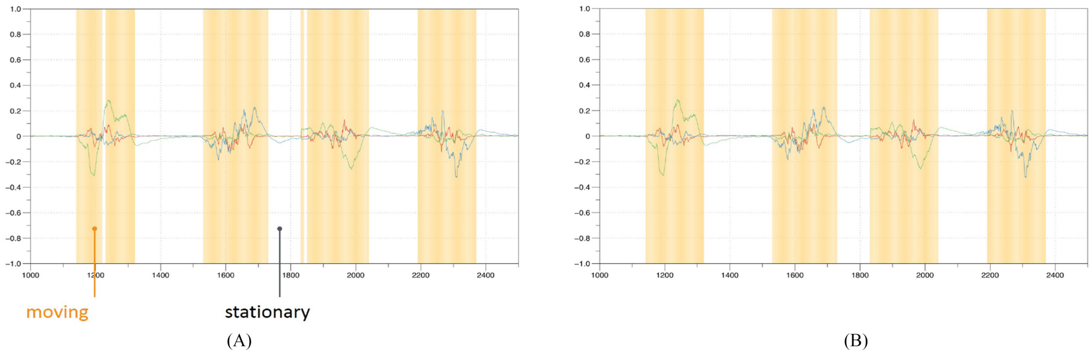


Fig. 8. (A) Classification results by LDA. (B) The smooth classification results after applying morphological operations.

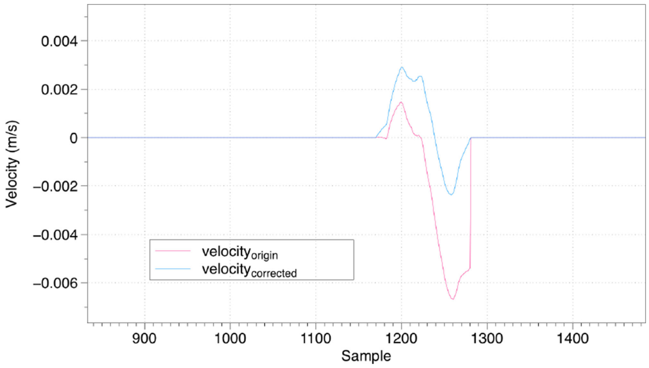


Fig. 9. The change of velocities with error and correction.

corrected velocity is then obtained by

$$offset = \frac{velocity(t_{end}) - velocity(t_{end} - 1)}{t_{end} - t_{begin}} \tag{12}$$

$$velocity_{correct}(t) = velocity(t) + (t - t_{begin} + 1) \times offset. \tag{13}$$

where t_{begin} is the time of entering a moving state and t_{end} is the time of finishing the current moving state. This method is similar to the zero velocity compensation (ZVC) method proposed by Yang [26].

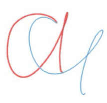
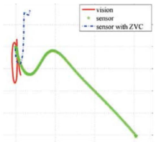
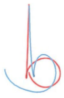
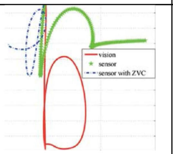
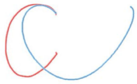
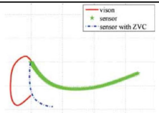
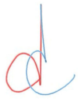
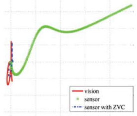
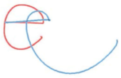
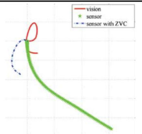
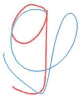
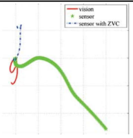
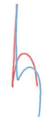
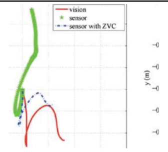
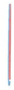
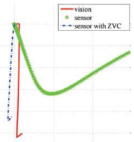
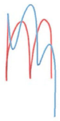
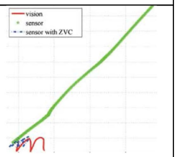
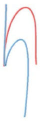
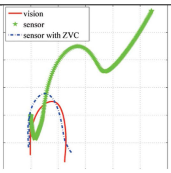
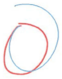
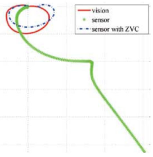
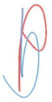
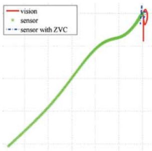
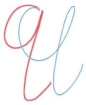
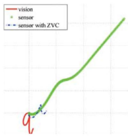
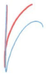
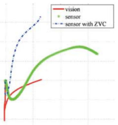
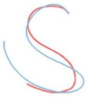
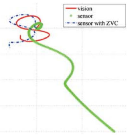
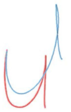
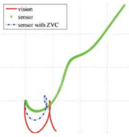
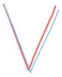
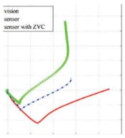
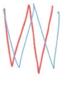
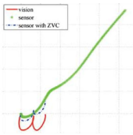
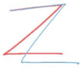
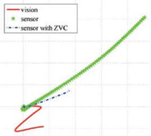
IV. EXPERIMENTAL RESULTS

To evaluate the proposed method, we used a low-cost 6-DoF IMU (GY-521 containing a MPU-6050 IC, which only costs 6 USD) to write lowercase letters from 'a' to 'z'. Since our method is designed for reconstructing a single trajectory, all letters were written in a single stroke. Y-Z plane is used for trajectory computation (cf. Fig. 5), and the user has to write with usual writing speed. Note that the user cannot write too slowly, otherwise the error drift problem will significantly affect the reconstruction result. A capacitive stylus was attached to the IMU sensor so that a tablet can be used to record the groundtruth of the written trajectory. In this section, our proposed method is validated by the following two experiments: 1) visualization of trajectory reconstruction; 2) recognition of trajectory reconstruction.

A. Visualization of Trajectory Reconstruction

To compare the trajectories reconstructed by our method with the ones reconstructed by the ZVC method [33], we visualize the trajectory reconstruction results. Each letter was recorded three times and we chose the one with least error distance to be shown in Table II. Euclidean distance is utilized to calculate the error distance between the groundtruth and the reconstructed trajectory. For the result of our proposed IMU-based method, the red line is the groundtruth of the trajectory and the blue line is the reconstructed trajectory. For the result of the ZVC

TABLE II
THE EXPERIMENTAL RESULTS OF OUR PROPOSED METHOD AND THE ZVC METHOD [33]

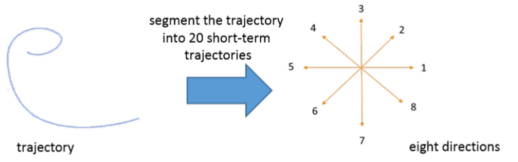
Proposed	ZVC	Proposed	ZVC	Proposed	ZVC
					
					
					
					
					
					
		🟦 Reconstruction 🟥 Groundtruth	🟦 Sensor with ZVC 🟥 Vision 🟩 Sensor		

For our proposed method, the red line is the groundtruth of the trajectory and the blue line is the reconstructed trajectory. For the ZVC method, the red line represents the vision-based reconstructed trajectory, which is taken as their groundtruth; the blue line and the green line represent the reconstructed trajectory with ZVC and without ZVC, respectively.

method, the red line is the trajectory obtained by using camera tracking technology (which can also be considered as the groundtruth), the green line is the result obtained by a baseline IMU-based method and the blue dot line is the result obtained by the ZVC method. Our method outperforms the ZVC method in most cases. We observed that the reconstructed trajectories of letter 'm' and 'n' are skewed, which might be caused by the acceleration generated by unconscious hand tremble during writing a stroke with large curvature. It consequently results in significantly influence of the reconstruction results. The difference between the groundtruth and the reconstructed trajectories might be caused by the inaccurate movement detection results. In the future, we will investigate more representative features that can help us to determine the exact durations of static/moving segments.

B. Recognition of the Reconstructed Trajectory

To evaluate our trajectory reconstruction method, we further applied two kinds of letter recognition approaches (i.e., sequence-based and image-based recognition) to check whether the reconstructed trajectory can be correctly recognized. Five subjects were asked to participate the experiment by writing a set of 19 letters which can be written in one stroke. Each letter was recorded ten times by each subject, and we used our trajectory reconstruction method to obtain the corresponding handwritten trajectory for each stroke of the recorded IMU signals. In total, there are 950 results of reconstructed trajectories in our dataset. We evaluated the performance of letter recognition by leave-one-out cross-validation method. All the signals are divided into five folds based on the subjects to perform 5-fold cross validation. The training set is composed of the



No. of direction	8	1	1	1	2	3	4	4	5	5	6	7	7	8	8	1	1	1	1	1
Fig. of direction	↘	→	→	→	↗	↑	↖	↖	←	←	↙	↓	↓	↘	↘	→	→	→	→	→

Fig. 10. An example of feature extraction for a handwritten letter ‘e’.

signals recorded by 4 subjects, and the signals recoded by the remaining subject are left as the testing set.

1) Sequence-Based Recognition: The features we used to classify the trajectory are based on the orientation of short-term trajectory displacement. For a trajectory composed of k sampled points $\{(x_0, y_0), (x_1, y_1), ..., (x_{k-1}, y_{k-1})\}$, we segment the trajectory into 20 short-term trajectories with equal sampled points and denote each short-term trajectory as $C(n)$, n=0, 1, ..., 19. Each short-term trajectory can be represented by $C(n)_{head}$ and $C(n)_{tail}$, which are the first point and the last point of this trajectory segment, respectively. To be more precise, the nth segment can be represented by

$$C(n)_{head} = (x_{\frac{k}{20}n}, y_{\frac{k}{20}n}),$$

$$C(n)_{tail} = (x_{\frac{k}{20}(n+1)-1}, y_{\frac{k}{20}(n+1)-1}). \tag{14}$$

Then, we classify the orientation of each short-term trajectory, i.e., $\theta(n)$, into one of the eight directions (as shown in Fig. 10) by

$$\begin{cases} 1 & : \quad 0^\circ < \theta(n) \leq 22.5^\circ, 337.5^\circ < \theta(n) \leq 360^\circ \\ 2 & : \quad 22.5^\circ < \theta(n) \leq 67.5^\circ \\ 3 & : \quad 67.5^\circ < \theta(n) \leq 112.5^\circ \\ 4 & : \quad 112.5^\circ < \theta(n) \leq 157.5^\circ \\ 5 & : \quad 157.5^\circ < \theta(n) \leq 202.5^\circ \\ 6 & : \quad 202.5^\circ < \theta(n) \leq 247.5^\circ \\ 7 & : \quad 247.5^\circ < \theta(n) \leq 292.5^\circ \\ 8 & : \quad 292.5^\circ < \theta(n) \leq 337.5^\circ, \end{cases} \tag{15}$$

where

$$\theta(n) = \tan^{-1} \frac{y_{\frac{k}{20}(n+1)-1} - y_{\frac{k}{20}n}}{x_{\frac{k}{20}(n+1)-1} - x_{\frac{k}{20}n}}. \tag{16}$$

A 20-dimensional feature vector describing the reconstructed trajectory is then used for letter recognition. Fig. 10 shows an example for feature extraction. DTW and HMM are then applied as the classifier, and Table III shows the average recognition accuracy can achieve 88.53% and 71.68%, respectively. Fig. 11 shows an error case, in which the trajectory reconstruction results of ‘c’ and ‘o’ are easily confusing.

2) Image-Based Recognition: We applied CNN in our image-based letter recognition approach by fine-tuning the CaffeNet model on the environment of 64-bit Windows 7 PC with a GEFORCE GTX 1060 GPU and 6GB RAM. The ImageNet dataset [34] was used to pre-train our network. The training data of handwritten letters collected from EMNIST dataset [35] was used to fine-tune the weights of our CNN network. Each image is warped to the size of 256*256 pixels and then a region of 224*224 pixels is cropped to be fed into networks. The base learning rate was set as 0.0001 for the lower layers, while the learning rate was multiplied by 10 in the last two fully-connected layers to speed up the adjustment of weights. The weight decay and momentum were fixed to 0.0005 and 0.9, respectively. As for the batch size, we set it to 115 for one iteration and the training epoch was set to 50. In this experiment, 950 images of reconstructed trajectories were used for testing. Table III shows that the average recognition accuracy can achieve 92.74%, which outperforms the two sequence-based recognition methods.

TABLE III
THE EXPERIMENTAL RESULTS OF HANDWRITTEN LETTER RECOGNITION

	DTW		HMM		CNN	
Letter	Accuracy (%)	Wrongly Recognized as	Accuracy (%)	Wrongly Recognized as	Accuracy (%)	Wrongly Recognized as
a	100		100		86	c, g, u
b	100		76	s, g	98	o
c	90	o	90	o	100	
d	92	n	42	b, n, q, s	88	e, l
e	98	c	72	a, o	100	
g	82	h	100		98	s
h	88	r	80	r	86	b, m, n
l	100		92	v	100	
m	100		70	p	98	h
n	98	d	84	b	74	h, m
o	80	c	86	e	100	
p	76	a, b, d	48	s, l	82	b
q	92	g	40	g	70	a, g
r	82	v	28	h, u, v, w	98	n
s	100		96	g	96	a, c
u	54	h	50	h, w	92	d
v	58	l, r,	44	l, s, u, w	100	
w	92	h	100		96	h
z	100		60	s	100	
Average	**88.53**		**71.47**		**92.74**	

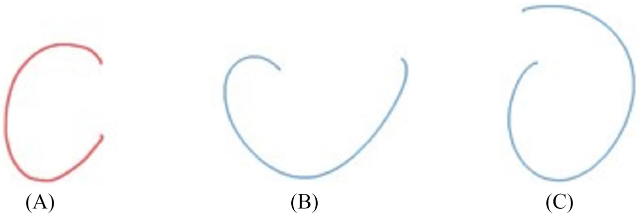


Fig. 11. The error case of recognition result between ‘c’ and ‘o’. (A) shows the groundtruth of ‘c’. (B) shows the reconstruction result of (A). (C) shows a trajectory of ‘o’, which looks quite similar as (B) and is then wrongly recognized as ‘c’.

V. CONCLUSIONS AND FUTURE WORK

This paper present a trajectory reconstruction method for a low-cost IMU equipped in the smartphone. We propose a reset switch method to reduce the accumulated error caused by the IMU sensor, and apply LDA to achieve more accurate movement detection which helps to generate the trajectory of the smartphone. The experiments show that our method can reconstruct better trajectory compared to the ZVC method. Furthermore, two sequence-based methods (i.e., DTW, HMM) and an image-based approach (i.e., CNN) were utilized to recognize the reconstructed letters, and the CNN approach was proved to have the best recognition accuracy on the reconstructed trajectories. Since the velocity integrated from the estimated accelerations is not exactly zero because of noises that can not be eliminated in the preprocessing step, it consequently results in serious error in displacement calculation when writing multi-stroke letters. In the future, we will try to utilize button (e.g., volume button of smartphone) as a trigger for recording each single stroke and resetting the velocity. Furthermore, in the process of our experiment, we also found that the acceleration caused by unconscious hand tremble of the user might significantly affect the handwriting reconstruction results. We will analyze signals in each axis and find the corresponding importance weighting for movement detection in each axis based on machine learning techniques to more accurately reconstruct the trajectory in the future.

References

[1] H. Tu, X. Ren, and S. Zhai, "A comparative evaluation of finger and pen stroke gestures," in *Proc. ACM SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 1287–1296.

[2] Wacom, Inkling, 2013. [Online]. Available: http://www.wacom.com/en/us/en/creative/inkling

[3] Livescribe, Echo smartpen, 2011. [Online]. Available: http://www.livescribe.com/en-us/smartpen/echo/

[4] Y. Zhao, "Applying time-differenced carrier phase in nondifferential GPS/IMU tightly coupled navigation systems to improve the positioning performance," *IEEE Trans. Vehicular Technol.*, vol. 66, no. 2, pp. 992–1003, Feb. 2017.

[5] A. Marquez, B. Tank, S. K. Meghani, S. Ahmed, and K. Tepe, "Accurate UWB and IMU based indoor localization for autonomous robots," in *Proc. 30th IEEE Canadian Conf. Electr. Comput. Eng.*, 2017, pp. 1–4.

[6] Y. Kim and Y. Li, "Human activity classification with transmission and reflection coefficients of on-body antennas through deep convolutional neural networks," *IEEE Trans. Antennas Propag.*, vol. 65, no. 5, pp. 2764–2768, May 2017.

[7] J. Hannink, T. Kautz, C. F. Pasluosta, K.-G. Gaßmann, J. Klucken, and B. M. Eskofier, "Sensor-based gait parameter extraction with deep convolutional neural networks," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 1, pp. 85–93, Jan. 2017.

[8] X. Li, Y. Zhang, I. Marsic, A. Sarcevic, and R. S. Burd, "Deep learning for RFID-based activity recognition," in *Proc. 14th ACM Conf. Embedded Netw. Sensor .ems*, 2016, pp. 164–175.

[9] C.-W. Yeh, T.-Y. Pan, and M.-C. Hu, "A sensor-based official basketball referee signals recognition system using deep belief networks," in *Proc. 23th Int. Conf. MultiMedia Model.*, 2017, pp. 565–575.

[10] X. Jin, O. Kanzo, and S. X. Bing, "A new positioning scheme for pen-like handwriting input devices," in *Proc. IEEE Consumer Commun. Netw. Conf.*, 2005, pp. 166–171.

[11] M. Sperber, M. Klinkigt, K. Kise, M. Iwamura, B. Adrian, and A. Dengel, "Handwriting reconstruction for a camera pen using random dot patterns," in *Proc. IEEE Int. Conf. Frontiers Handwriting Recog.*, 2010, pp. 160–165.

[12] M. Carminati, G. Ferrari, R. Grassetti, and M. Sampietro, "Real-time data fusion and MEMS sensors fault detection in an aircraft emergency attitude unit based on Kalman filtering," *IEEE Sensors J.*, vol. 12, no. 10, pp. 2984–2992, Oct. 2012.

[13] I. Frosio, F. Pedersini, and N. A. Borghese, "Autocalibration of triaxial MEMS accelerometers with automatic sensor model selection," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2100–2108, Jun. 2012.

[14] M. Sipos, P. Paces, J. Rohac, and P. Novacek, "Analyses of triaxial accelerometer calibration algorithms," *IEEE Sensors J.*, vol. 12, no. 5, pp. 1157–1165, May 2012.

[15] A. Sabato, C. Niezrecki, and G. Fortino, "Wireless MEMS-based accelerometer sensor boards for structural vibration monitoring: a review," *IEEE Sensors J.*, vol. 17, no. 2, pp. 226–235, Jan. 2017.

[16] L. Tong, Q. Song, Y. Ge, and M. Liu, "HMM-based human fall detection and prediction method using tri-axial accelerometer," *IEEE Sensors J.*, vol. 13, no. 5, pp. 1849–1856, May 2013.

[17] R. M. Hagem, S. G. O'Keefe, T. Fickenscher, and D. V. Thiel, "Self contained adaptable optical wireless communications system for stroke rate during swimming," *IEEE Sensors J.*, vol. 13, no. 8, pp. 3144–3151, Aug. 2013.

[18] S. Zihajehzadeh, T. J. Lee, J. K. Lee, R. Hoskinson, and E. J. Park, "Integration of MEMS inertial and pressure sensors for vertical trajectory determination," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 3, pp. 804–814, Mar. 2015.

[19] S. Zihajehzadeh and E. J. Park, "Regression model-based walking speed estimation using wrist-worn inertial sensor," *PLoS One*, vol. 11, no. 10, 2016, Art. no. e0165211.

[20] R. Xie, X. Sun, X. Xia, and J. Cao, "Similarity matching-based extensible hand gesture recognition," *IEEE Sensors J.*, vol. 15, no. 6, pp. 3475–3483, Jun. 2015.

[21] R. Xie and J. Cao, "Accelerometer-based hand gesture recognition by neural network and similarity matching," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4537–4545, Jun. 2017.

[22] 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.

[23] J.-S. Wang, Y.-L. Hsu, and C.-L. Chu, "Online handwriting recognition using an accelerometer-based pen device," in *Proc. IEEE Int. Conf. Adv. Comput. Sci. Eng.*, 2013, pp. 229–232.

[24] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition," *IEEE Sensors J.*, vol. 15, no. 1, pp. 154–163, Jan. 2015.

[25] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proc. 9th ACM Int. Conf. Mobile Syst., Appl. Services*, 2011, pp. 15–28.

[26] J. Yang *et al.*, "Analysis and compensation of errors in the input device based on inertial sensors," in *Proc. IEEE Int. Conf. Inform. Technol.: Coding Comput.*, vol. 2, 2004, pp. 790–796.

[27] J.-S. Wang, Y.-L. Hsu, and J.-N. Liu, "An inertial-measurement-unit-based pen with a trajectory reconstruction algorithm and its applications," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3508–3521, Oct. 2010.

[28] M. Sepahvand, F. Abdali-Mohammadi, and F. Mardukhi, "Evolutionary metric-learning-based recognition algorithm for online isolated Persian/Arabic characters, reconstructed using inertial pen signals," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2872–2884, Sep. 2017.

[29] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recog.*, vol. 48, no. 10, pp. 3102–3112, 2015.

[30] B. Du *et al.*, "Exploring representativeness and informativeness for active learning," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 14–26, Jan. 2017.

[31] M. Pedley, "High precision calibration of a three-axis accelerometer," Freescale Semiconductor, Inc., Austin, TX, USA, Application Note AN4399, 2015.

[32] J. S. Hunter, "The exponentially weighted moving average," *J. Quality Technol.*, vol. 18, no. 4, pp. 203–210, 1986.

[33] Z. Dong, U. C. Wejinya, and W. J. Li, "An optical-tracking calibration method for MEMS-based digital writing instrument," *IEEE Sensors J.*, vol. 10, no. 10, pp. 1543–1551, Oct. 2010.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[35] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," 2017. [Online]. Available: arXiv:1702.05373

Tse-Yu Pan received the B.S. degree in computer science and information engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2013, and is currently working toward the Ph.D. degree in computer science and information engineering at National Cheng Kung University, Tainan, Taiwan. He is a recipient of the Doctoral Fellowship from Pan Wen Yuan Foundation and the Student Travel Grant from ACM Multimedia Conference in 2016 and 2017, respectively.

His research interests include digital signal processing, pattern recognition, computer vision, and multimedia information system.

Chih-Hsuan Kuo received the B.S. and M.S. degrees in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2012 and 2014, respectively. His research interest include human–computer interaction.

Hou-Tim Liu received the B.S. and M.S. degrees in computer science information engineering from the National Cheng Kung University, Tainan, Taiwan, in 2014 and 2016, respectively. His research interest include image processing, machine learning, and handwritten language identification.

Min-Chun Hu is also known as Min-Chun Tien and Ming-Chun Tien. She received the B.S. and M.S. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively, and the Ph.D. degree from the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan, in 2011. She was a Postdoctoral Research Fellow at Research Center for Information Technology Innovation, Academia Sinica, from 2011 to 2012. She is an Assistant Professor at the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan. Her research interests include digital signal processing, multimedia content analysis, pattern recognition, computer vision, computer graphics, virtual reality, and augmented reality. She was a recipient of the Exploration Research Award from Pan Wen Yuan Foundation and the Outstanding Young Researcher Award from the Computer Society of the Republic Of China in 2015 and 2017, respectively.