Fine-Grained and Real-Time Gesture Recognition by Using IMU Sensors

Dian Zhang, *Member, IEEE*, Zexiong Liao, Wen Xie, Xiaofeng Wu,
Haoran Xie, *Senior Member, IEEE*, Jiang Xiao, and Landu Jiang

Abstract—Gesture recognition by using Inertial Measurement Unit (IMU) sensors plays an important role in various Internet of Things (IOT) applications, e.g., smart home, intelligent medical system and so on. Traditional technologies usually utilize machine learning algorithms to train different gestures during the offline phase, then recognize the gesture during the online phase. However, such technologies cannot recognize these gestures without prior training. Even for the same gesture, with different gesture amplitude may result in unsuccessful recognition. Also if we change the person to perform the same gesture, the algorithms fails. In order to overcome these drawbacks, we propose an approach, which will be able to track the human body motion in real-time and also recognize complicated gestures. It utilizes the accelerometer information and proposes comprehensive localization algorithms for each deployed sensor attached on the human body. Then, it takes the correlation and limitation among body parts into account to recognize the gesture. Our experiments results show that, the successful recognition rate of our algorithm is 100%. Furthermore, any part of the human body can be well tracked, the tracking accuracy can reach $0.06m$.

Index Terms—IMU, gesture recognition, motion tracking

1 Introduction

Gesture recognition using wireless technologies plays an important role in many applications. For example, various Virtual Reality (VR) applications and products require to show the human's gesture in the virtual world [1], [2]. In the smart home applications [3], [4], the human gestures recognition is quite useful to decide how to control these intelligent devices. In the intelligent medical system [5], [6], users may utilize gesture recognition to discover the patient motion behavior and their intention easily. Compared to traditional vision-based technologies [2], [7], [8]. Wireless technologies are not limited by line of sight or light condition of the devices. Furthermore, user privacy can be protected using such technologies. Thus, gesture recognition using wireless technologies attract the attention of many researchers.

Among current gesture recognition based on wireless technologies, some utilize WiFi or other device to detect the signals influenced by the target [9], [10], [11]. But the signals are easily affected by the surroundings, causing the accuracy to vary the accuracy vary easily. Some other technologies adopt wearable devices, e.g., Inertial Measurement Unit (IMU) sensors. However, such gesture recognition algorithms usually utilize machine learning algorithms to train some data set during the offline phase, then run it during the online phase [12], [13], [14]. These method can only recognize common human gestures. If new gesture occurs, these recognition algorithms tend to fail easily. Even for the same gesture that has already been trained, different gesture amplitude may result in failure to recognize. Also if we change the person to perform the same gesture, the algorithms will fail easily.

In order to overcome these drawbacks introduced earlier, we propose an approach base on IMU sensors, which will be able to recognize various human gestures having different amplitude precisely in real time, without having to train during the offline phase. The added advantage is that complicated gestures can also be recognized.

Our basic idea is to utilize the accelerometer information to localize each deployed sensor on the human body. Our algorithms not only can refine the hardware linear distortion, but also can eliminate the impact of the gravity and hardware differences. Furthermore, the correlation between the deployed sensors on human body also utilized to refine the tracking results. As a result, we may easily recognize different human gestures and track human body motion. Even for the same human body gestures with different amplitudes, our algorithm is able to recognize and differentiate them easily.

To sum up, the contribution of our paper is listed as below.

1) Our proposed approach can precisely recognize the human gestures with different amplitude and also track body motion;

- *Dian Zhang, Zexiong Liao, Wen Xie, and Landu Jiang are with the Shenzhen University, Shenzhen 518060, China. E-mail: serena.dian@gmail.com, {1810273017, 1810273027}@email.szu.edu.cn, landu.jiang@mail.mcgill.ca.*
- *Xiaofeng Wu is with the Shenzhen Research Institute of Big Data, Shenzhen 518172, China. E-mail: 867746816@qq.com.*
- *Haoran Xie is with the Lingnan University, Hong Kong. E-mail: hrxie@ln.edu.hk.*
- *Jiang Xiao is with the Huazhong University of Science and Technology, Wuhan, Hubei 430074, China. E-mail: jiangxiao@hust.edu.cn.*

Manuscript received 20 November 2020; revised 24 September 2021; accepted 28 September 2021. Date of publication 30 November 2021; date of current version 6 March 2023.
This work was supported in part by the NSFC under Grant 61872247, in part by the Shenzhen Peacock Talent under Grant 827-000175, and in part by the Guangdong Natural Science Funds under Grant 2019A1515011064.
(Corresponding authors: Jiang Xiao and Landu Jiang.)
Digital Object Identifier no. 10.1109/TMC.2021.3120475

2) Compared to traditional technologies, our algorithms can be performed without prior training, making it to be widely used in real scenarios, like virtual reality or interactive game;
3) We utilize the limitation of the body motion and the correlation among the body parts to improve the accuracy of the gesture recognition and motion tracking.

Our experiments are based on IMU sensors, each includes a 3-axis accelerometer, a 3-axis gyroscope and 3-axis Compass. These sensors are deployed on different parts of the human bodies. After 40 rounds of different gesture tests, experiment results show that the successfully recognition rate of our algorithm is 100%. Furthermore, the average localization error of all the deployed sensors is only $0.06\,m$.

The rest of this paper is organized as follows. We will introduce the related work in the Section 2. Section 3 contains details of our methodology. The implementation of our experiment and accuracy of the recognition will be presented at Section 4. Finally, we will conclude this paper and point our future work.

2 Related Work

2.1 Gesture Recognition Technology

Tradition gesture recognition technologies basically can be classified into 4 categories: vision based technologies, radio technologies, wearable device technologies and hybrid technologies.

1. Vision based technologies utilize the cameras to capture the gesture by applying various image processing algorithms [15], [16], [17]. Juang [18] is able to classify the human body gesture using an interval type-2 neural fuzzy classifier. Ohn-Bar [19] can recognize driver hand gestures by employing a combined RGB and depth descriptor. Brulin [20] leveraged fuzzy logic to recognize 4 static gestures for elderly. Ren [21] utilized Kinect sensors and proposed a Finger-Earth Mover$\bar{a}$s Distance (FEMD) metric to measure the difference between different hand shapes, it can reach a high accuracy based on a dataset of 10 gestures. Ju *et al.* [22] purposed a integrative framework to segment hand gestures using the Kinect device. Kevin *et al.* [23] have developed hand gestures recognition based on Kinect camera using Histogram of Oriented Gradient (HOG) and Dynamic Time Warping (DTW) methods. There are some work utilize Leap Motion controller to recognize gesture [24], [25], [26]. These works usually require light for the environment. Furthermore, the camera has a limited covering area. The privacy problem is also a concern.

2. Radio technologies are popular since they have no light limitation and privacy concern. In such technologies, usually a number of various wireless devices are utilized, e.g., the WiFi devices [9], [10]. Activity recognition and monitoring system (CARM) [9] is able to recognize device-free human target activity based on the Channel State Information (CSI) signals difference caused by the target. In the CARM, Hidden Markov Model (HMM) is used to build the activity model. He [10] proposed a device-free gesture recognition system based on WiFi infrastructure and devices. Wang *et al.* [27] utilized the mmWave signals to realize gesture recognition. However, when signals are blocked by obstacles (Non-Line-of-Sight signals [28]), the accuracy of recognition will be reduced. WiTrack [29] can track the 3D motion of a user from the radio signals reflected by the human body. But it only can provide coarse tracking of body parts. Pantomime [30] can realize gesture recognition through millimeter-wave radio frequency signals. But it has lower accuracy if gestures are performed along the Z axis and the gestures should be performed at slow speed. Widar3.0 [31] is a Wi-Fi based zero-effort cross-domain gesture recognition system. But different user heights will influence the gesture recognition rate.

3. A large number of works aim to recognize target gestures using the wearable devices. Among them, technologies based on IMU devices [12], [32], [33], [34] are most popular. The IMU sensors include an accelerometer and a gyroscope. Jerome [35] utilized a neural network algorithm to recognize gestures wearing smart gloves, but it is only limited to static gestures. Xu [32] utilized the IMU sensors and proposed an automatic gesture segmentation algorithmm, which is developed to identify individual gestures in a sequence to recognize seven hand gestures. Wu [12] utilized arm orientation limitation and joint angle to refine the gesture model. However, the above technologies can only estimate the likelihood of a number of predefined gestures and cannot recognize the same gestures with different amplitude. S.H.P [36] utilized a micro-inertial sensor and a three-axis magnetometer to propose a tracking tool system. This system mainly correct the position of the tool. Gupta [37] proposed a continuous gesture recognition algorithm, which mainly uses the IMU sensors on the mobile phone to compress and encode each gesture. It aims to mark the start and end of each gesture, then achieves a continuous gesture recognition. However, this algorithm only recognizes predefined gestures. Yun's work [38] is able to track the human body motion in real-time by using angular rate sensor and accelerometer sensor. However, it utilized the angular motion of the body (limb) to recognize simple gestures, but it cannot recognize complicated gestures, e.g., having both displacement and rotation. The continuous hand gesture recognition methodology [37] utilized the IMU sensors. This methodology detected the start and end points of meaningful gesture segments, then used the DTW algorithm to identify a gesture. However, it can only recognize gestures that have been categorized in advance.

There are also some technologies based on other wearable device proposed in recent years. Watanabe *et al.* [39] utilized sound wave to recognize gestures and proposed a method to improve ultrasound-based gesture recognition by attaching a cover to the microphone. Zhang *et al.* [40] proposed a gesture input technique using Photoplethysmography (PPG) signal that optical heart-rate sensors capture. They utilized optical sensors on the off-the-shelf wearable devices to recognise gestures. Becker *et al.* [41] utilized a wireless electromyography (EMG) armband to classify the finger touches and estimates their force. Zhang *et al.* [42] proposed a novel gesture recognition framework to learn intrinsic spatial and temporal features of surface electromyography (sEMG). But these methods are either with higher cost or inconvenient to use.

4. Some researchers used hybrid technologies in gesture recognition. Chan *et al.* [43] designed a hand tracking glove. It employed sensing data from three different sensors, which

includes a camera, an IMU and flex sensors. Besides, the Kalman filter is applied to stabilize the pose acquired. These technologies requires various devices to be deployed and are inconvenient to use in practice.

2.2 Gesture Recognition Algorithms

There are huge amount of work [44], [45], [46], [47], [48], [49], [50], [51] leveraging machine learning algorithms to recognize the human gestures. Zhu and Sheng [48] utilized the Neural Network and Hidden Markov Model to recognize gesture. Zhao [49] proposed a real-time gesture recognition system for head-mounted devices. This system mainly trained the collected head motion information to establish a hidden Markov Model, then recognized the gesture in real time. Wang and Chuang [46] utilizes Linear Discriminant Analysis (LDA) to extract feature and employ Probabilistic Neural Network (PNN) to recognize gestures. Wu and Sun [50] utilized the muscle sensors and inertial sensors to collect the gesture data of sign language, then used the Support Vector Machine (SVM), Naive Bayes (NB) and K-Nearest Neighbor (KNN) algorithm to classify the gestures. Bhuyan [51] utilized the gyroscope and accelerometer to predict the direction of rotation of the arm, then uses Kalman filtering to find the exact position of the arm. Wang *et al.* [27] developed tow Generative Adversarial Networks (GANs), i.e., SS-GAN and ST-GAN, to improve human gesture recognition accuracy. Zhang *et al.* [42] designed a convolutional recurrent neural network (CRNN) architecture to identify hand gestures. Gochoo *et al.* [52] proposed an IoT-based privacy-preserving yoga posture recognition system employing a deep convolutional neural network (DCNN).

However, these machine learning algorithms are only able to recognize gestures already trained during the offline phase. If a different gesture occurs, or the target human has been replaced, or even the same person performs same gesture but with different amplitude, these methodologies usually fail to recognize them. Therefore, machine learning algorithms only can recognize limited number of predefined gestures. Especially, their training cost is high.

In our work, we can accurately recognize complicate gestures in real time, even if gestures are similar but with different amplitude. Compared to machine learning algorithms, our algorithms do not require high computational costs and can be performed without prior training.

3 Methodology

In our approach, we first eliminate the noise and linear distortion of the hardware devices, then eliminate the impact of gravity on such data. In the following, we propose a comprehensive algorithm to track human gesture in real time, which can eliminate the hardware sensitivity difference and effectively utilize limitation of body motion to improve the tracking accuracy dramatically. Finally, we propose a recognition algorithm to recognize human body gesture.

3.1 Gyroscope Noise Reduction

The gyroscope sensor mainly consists of three kinds of errors. The first error is the high-frequency noise. The second error is the linear distortion of the hardware caused by the welding inaccuracy, plane curvature and roughness, as well as circuit board processing accuracy. The third error is the zero draft. In order to remove the first error, the low-pass filter algorithm [53] is utilized to eliminate the high frequency noise. Fig. 1 shows the second error. We see that, the x, y and z axis in theory are completely orthogonal to each other in the ideal state. But in practice, acceleration measurement axis may not be in this ideal case. That is, the ideal conditions of the three-axis and the actual sensor do not entirely coincide. It is known as the sensor device error.

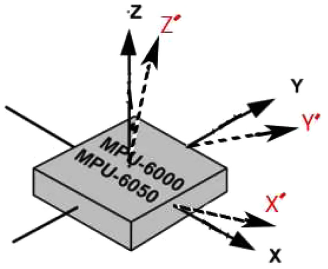


Fig. 1. Hardware linear distortion. x, y and z axis are the ideal state, while $x^{'}$, $y^{'}$ and $z^{'}$ are the actual state.

In order to eliminate second error, for each accelerometer sensor, we use the following linear fit equation to recover the accelerometer value a_{x_T}, a_{y_T} and a_{z_T} in the vector coordination (hardware coordinate) system

$$\begin{aligned} a_{x_T} &= k_x * a_x + b_x \\ a_{y_T} &= k_y * a_y + b_y \\ a_{z_T} &= k_z * a_z + b_z. \end{aligned} \tag{1}$$

In the above equation, a_{x_T}, a_{y_T}, a_{z_T} are the true value of the accelerometer along the x axis, y axis and z axis, respectively. a_x, a_y, a_z are the measured accelerometer data along the x, y and z axis, respectively. k_x and b_x, k_y and b_y, k_z and b_z are the correction factors of the accelerometer along the x, y and z axis, respectively, when the sensor stays static.

Taking the x axis as an example, we can use the least square method to determine the parameters k_x and b_x. In the above equation, a_{xt} is the real value, while a_x is the measured value. In order to minimize the weighted average sum of the observed values, we only need to minimize the value of the following equation. Take partial derivatives of k_x and b_x of the above formula, then

$$\begin{aligned} &\frac{\partial}{\partial b_x}\sum_{i=1}^{N}[a_{xti} - (b_x + k_x a_{xi})]^2 \\ &= -2\sum_{i=1}^{N}[a_{xti} - b_x - k_x a_{xi}] = 0 \end{aligned} \tag{2}$$

$$\begin{aligned} &\frac{\partial}{\partial k_x}\sum_{i=1}^{N}[a_{xti} - (b_x + k_x a_{xi})]^2 \\ &= -2\sum_{i=1}^{N}[a_{xti} - b_x + (k_x a_{xi})] = 0. \end{aligned} \tag{3}$$

After finishing to

$$\begin{cases} b_x N + k_x \sum_{i=1}^{N} a_{xi} = \sum_{i=1}^{N} a_{xti} \\ b_x \sum_{i=1}^{N} a_{xi} + k_x \sum_{i=1}^{N} a_{xti}^2 = \sum_{i=1}^{N} a_{xti} a_{xi} \end{cases}. \tag{4}$$

TABLE 1
Notations

Notation	Description
a_x, a_y, a_z	a_x, a_y and a_z respectively represent the acceleration data of x, y and z axis output by IMU inertial sensor.
g_x, g_y, g_z	g_x, g_y and g_z respectively represent the angular velocity data of x, y and z axis output by IMU inertial sensor.
C_b^w, C_w^b	C_b^w represents the transformation matrix from the carrier coordinate system to the world coordinate system, C_w^b represents the transformation matrix from the world coordinate system to the carrier coordinate system. They are inverse matrices.
(W_x, W_y, W_z)	(W_x, W_y, W_z) is the initial coordinate position of the sensor.
$(W_x^{'}, W_y^{'}, W_z^{'})$	$(W_x^{'}, W_y^{'}, W_z^{'})$ is the initial coordinate position after the sensor is moved.
α, β, γ	α is the pitch around the x-axis. β is the yaw around the y-axis. γ is the roll angle around the z-axis.
$a_1, a_2, a_3, \ldots, a_n$	$a_1, a_2, a_3, \ldots, a_n$ refer to the received acceleration sequence, and a_i refers to the acceleration data received at time $i \times \triangle t$.
l	l represents the length of a part of the human body.

Solve the above equation to get

$$k_x = \frac{N\left(\sum_{i=1}^{N} a_{xi}a_{xti}\right) - \left(\sum_{i=1}^{N} a_{xi}\right)\left(\sum_{i=1}^{N} a_{xti}\right)}{N\left(\sum_{i=1}^{N} a_{xi}^2\right) - \left(\sum_{i=1}^{N} a_{xi}\right)^2} \tag{5}$$

$$b_x = \frac{\left(\sum_{i=1}^{N} a_{xi}^2\right)\left(\sum_{i=1}^{N} a_{xti}\right) - \left(\sum_{i=1}^{N} a_{xi}\right)\left(\sum_{i=1}^{N} a_{xi}a_{xti}\right)}{N\left(\sum_{i=1}^{N} a_{xi}^2\right) - \left(\sum_{i=1}^{N} a_{xi}\right)^2}. \tag{6}$$

In the same way, we can also calculate the correction factors k_y, b_y and k_z, b_z of y axis and z axis respectively, so as to eliminate the influence of mechanical error on acceleration.

Zero drift [54] means that in the static situation, the output data is not equal to zero. Therefore, we use the difference between the actual measured value and the ideal value in the static case to replace the gyroscope data. g_x, g_y, and g_z are the actual data along x, y and z axis output by the gyroscope respectively, while g_{xt}, g_{yt} and g_{zt} are the data along x, y and z axis respectively in the ideal state, then

$$\begin{aligned} g_{xt} &= g_x + c_x \\ g_{yt} &= g_y + c_y \\ g_{zt} &= g_z + c_z. \end{aligned} \tag{7}$$

In the above formula, c_x, c_y and c_z are the static compensation quantities of x, y and z axes respectively. The sensor is placed in an arbitrary position, and the average value of q sets of gyroscope data of x, y and z axes are c_x, c_y and c_z, respectively. In this way, the gyroscope data of the MEMS sensor can be corrected.

Some important notations that will be used in this paper are listed in Table 1.

3.2 Noise Reduction of Acceleration Sensor

The data noise of the acceleration sensor mainly comes from two aspects. The first is the error caused by the mechanical imperfection of the hardware. The second is the influence of the acceleration due to gravity that always exists in space. To eliminate the first error, we use the linear distortion algorithm mentioned in the previous section. The second error is that g will have an impact on the z-axis of the world coordinate system. However, the obtained sensor will transmit the sensing data based on the vector coordinate (hardware coordinate) system. Therefore, we should transform the acceleration data from the vector coordinate (hardware coordinate) system to the world coordinate system.

The gyroscope sensor describes how target object rotates. In Euler angle description, suppose ψ, θ and ϕ is the target rotation angle of z, y and x, representing Yaw, Pitch and Roll, respectively. According to [55], we can calculate the transformation matrix C_b^w from vector coordinate to world coordinate as shown below:

$$C_b^w = \begin{bmatrix} \cos\psi\cos\theta\cos\phi & \sin\psi\cos\theta & -\sin\theta \\ \begin{matrix}(\cos\psi\sin\theta\sin\phi - \\ \sin\psi\cos\phi)\end{matrix} & \begin{matrix}(\sin\psi\sin\theta\sin\phi + \\ \cos\psi\cos\phi)\end{matrix} & \cos\theta\sin\phi \\ \begin{matrix}(\cos\psi\sin\theta\cos\phi + \\ \sin\psi\sin\phi)\end{matrix} & \begin{matrix}(\sin\psi\sin\theta\cos\phi - \\ \cos\psi\sin\phi)\end{matrix} & \cos\theta\cos\phi \end{bmatrix}. \tag{8}$$

Suppose in the hardware coordinate system, the acceleration data matrix from the accelerometer sensor is as follows:

$$a_w = \begin{bmatrix} a_{wx} & a_{wy} & a_{wz} \end{bmatrix}^T. \tag{9}$$

In the absolute world coordinate system, the acceleration data can be expressed as shown below:

$$a_b = [C_b^w] \cdot [a_w] = \begin{bmatrix} a_{bx} & a_{by} & a_{bz} \end{bmatrix}^T. \tag{10}$$

Since the impact of gravity affect only in the z axis of the world coordinate system, we only need to eliminate the g value along the z axis, that is

$$a_{b_T} = \begin{bmatrix} a_{bx} & a_{by} & a_{bz} - g \end{bmatrix}. \tag{11}$$

Hence, the impact of $|g|$ will be eliminated.

3.3 Tracking Algorithm of the Body Motion

After the previous pre-processing procedure, we utilize the following algorithm to track the body motion. Basically, our tracking algorithm consists of 4 sub-algorithms, which are performed along the sequence. First, a data noise elimination algorithm is proposed in Section 3.3.1. Second, we put forward a tracking algorithm for each wearable device in

Section 3.3.2. Such algorithm is able to calculate the coordinates of each device. These coordinates will form a trace of such device. Third, we use a hardware difference elimination algorithm to refine the tracking accuracy in Section 3.3.3. At last, we leverage the correlation and limitation among body parts to optimize the tracking accuracy in Section 3.3.4.

3.3.1 Elimination of Data Noise

Suppose each sensor will transmit its acceleration data every time interval $\triangle t$, and n is the total number of transmitted packets, we may obtain acceleration data array a_n in theory. In practice, if a_i $(1 < i < n)$ is not received by the receiver, we need to insert an interpolation value in the array as follows:

$$a_i = (a_{i+1} + a_{i-1})/2. \tag{12}$$

In particular, when successive values are not received, we use the following formula to insert values:

$$a_{k_j} = \frac{a_{end} - a_{start}}{k+1} * j + a_{start}. \tag{13}$$

In the above formula, a_{k_j} means that the jth value in the k continuously missing values. a_{start} and a_{end} represents the first and the last of the continuously missing values respectively.

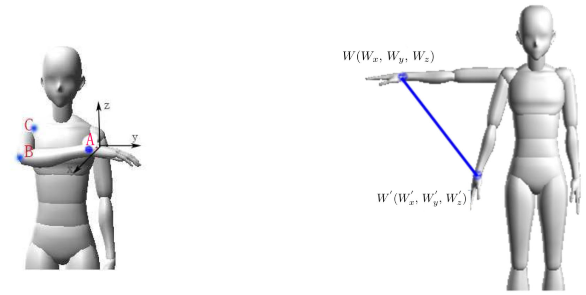
In the following, we utilize the five-spot triple smoothing method [56] to reduce the high frequency random noise. Suppose there are n number of received acceleration data represented by a data array a_n, within a sampling time interval $\triangle t$, and let a_{i-2}, a_{i-1}, a_i, a_{i+1} and a_{i+2} be any five continuous received acceleration data, they can be used to recalculated as follows:

$$\begin{aligned}
\overline{a}_{i-2} &= \frac{1}{70}(69a_{i-2} + 4a_{i-1} - 6a_i + 46a_{i+1} - a_{i+2}) \\
\overline{a}_{i-1} &= \frac{1}{35}(2a_{i-2} + 27a_{i-1} + 12a_i - 8a_{i+1} + 2a_{i+2}) \\
\overline{a}_{i} &= \frac{1}{35}(-3a_{i-2} + 12a_{i-1} + 17a_i + 12a_{i+1} - 3a_{i+2}) \\
\overline{a}_{i+1} &= \frac{1}{35}(2a_{i-2} - 8a_{i-1} + 12a_i + 27a_{i+1} + 2a_{i+2}) \\
\overline{a}_{i+2} &= \frac{1}{70}(-a_{i-2} + 4a_{i-1} - 4a_i + 4a_{i+1} + 69a_{i+2}).
\end{aligned} \tag{14}$$

In the above formula, $\overline{a}_{i-2}$, $\overline{a}_{i-1}$, $\overline{a}_i$, $\overline{a}_{i+1}$, $\overline{a}_{i+2}$ means the refine result respectively. The coefficients in the equations are refined by [56]. The initial a_0 and a_1 may be estimated from the first and second equation. The final a_{n-1} and a_n may be recalculated from the fourth and fifth equation. Other a_i can be derived from the third equation.

3.3.2 Basic Motion Tracking Algorithm for Each Wearable Device

In the previous subsections, we have already finished refining the hardware linear distortion, eliminating the impact of gravity and noise. Therefore, in this subsection, we may utilize the processed acceleration data to accurately calculate the displacement of each wearable device and derive the gesture in real time.



(a) Deployment example. The 3 wearable sensors A, B, C are put on the human wrist, elbow and shoulder, respectively.

(b) The displacement.

Fig. 2. (a) Deployment example. The 3 wearable sensors A, B, C are put on the human wrist, elbow and shoulder, respectively. (b) The displacement.

In our first attempt, we leverage simpson double integral [57] to calculate the displacement of each wearable device as follows:

$$\begin{aligned}
v_i &= v_{i-1} + ((a_{i-1} + 4a_i + a_{i+1})/6) \cdot \triangle t \\
s_i &= s_{i-1} + ((v_{i-1} + 4v_i + v_{i+1})/6) \cdot \triangle t,
\end{aligned} \tag{15}$$

where v_i is the velocity of the target sensor at time t_i $(i \in (1, n)$, n is the number of received acceleration data), s_i is the total accumulate displacement of the target sensor at time t_i, $\triangle t$ is the time interval of sampling. s_n is the total displacement after receiving n number of acceleration data.

In our work, the sensors are placed on the joints of the human body. Suppose that a certain fixed point of the body is the origin, then according to the relationship between the human joints, we can get the initial coordinates of the sensor at other joints. As shown in Fig. 2b, assume that there is a wearable sensing device W, let its initial coordinate be $W(W_x, W_y, W_z)$, then the coordinate of such device after moving is expressed as $W'(W'_x, W'_y, W'_z)$, which can be further expressed as

$$\begin{aligned}
W'_x &= W_x + s_{nx} \\
W'_y &= W_y + s_{ny} \\
W'_z &= W_z + s_{nz},
\end{aligned} \tag{16}$$

where s_{nx}, s_{ny} and s_{nz} are the integration results s_n mapping to x, y and z axis, respectively.

3.3.3 Improvement Method 1: Eliminate Hardware Difference on Tracking Accuracy

There are sensing variance among different wearable sensor devices. Even for the same sensor, the sensitivity of accelerometer along x, y and z axis are also different. Therefore, in order to eliminate such difference, for each sensing device, we calculate a correction factor u_p and u_n for the positive direction and negative direction along each axis. Taking the x axis as an example, u_{xp} and u_{xn} are the results u_p and u_n mapping to position and negative direction of x axis, respectively. It can be represented by the following equation:

$$\begin{aligned}
u_{xp} &= (d_{cxp} - d_{rxp})/d_{rxp} \\
u_{xn} &= (d_{cxn} - d_{rxn})/d_{rxn},
\end{aligned} \tag{17}$$

where d_r represents the real displacement of the wearable sensing device. d_r is obtained by manual measurement. d_c is the calculated displacement by the basic tracking algorithm introduced in the last subsection. In real application, we may perform a number of tests (e.g.,10), and utilize the average calculated displacement to represent d_c. d_{rxp} and d_{cxp} are d_r and d_c mapping to positive direction of x axis. d_{rxn} and d_{cxn} are d_r and d_c mapping to the negative direction of x axis.

Therefore, the displacement of x axis can be refined by u_{xp} and u_{xn} as follows:

$$\begin{aligned} s_{uxp} &= s_{nxp}/(1+u_{xp}) \\ s_{uxn} &= s_{nxn}/(1+u_{xn}), \end{aligned} \tag{18}$$

here s_{nxp} and s_{nxn} are mapping to the positive direction calculated displacement and negative direction calculated displacement of s_{nx}, respectively. Assume that there is a wearable sensing device M, its initial coordinate is $M(M_x$, M_y, $M_z)$, then the coordinate of such device after being refined by correction factor is expressed as $M^{'}(M^{'}_x$, $M^{'}_y$, $M^{'}_z)$, the x coordinate can be expressed as

$$M^{'}_x = M_x + s_{ux}, \tag{19}$$

here s_{ux} value determined by the following formula:

$$s_{ux} = \begin{cases} s_{uxp} & s_{nx} > 0 \\ s_{uxn} & s_{nx} <= 0 \end{cases}. \tag{20}$$

The correction factors of the other axis are calculated following the same rule.

3.3.4 Improvement Method 2: Enhance the Tracking Accuracy by Using the Correlation and Limitation Among Body Parts

Since each wearable device is fixed on a certain position of the human body, its moving trajectory will be in accordance with the common human motion. Therefore, we can take advantage of this correlation and limitation among body parts to optimize the tracking results.

We use the following criteria to determine whether to use the improvement method

$$M_{distance} = \sqrt{(M_x - M^{'}_x)^2 + (M_y - M^{'}_y)^2 + (M_z - M^{'}_z)^2} \tag{21}$$

$$G_i = \begin{cases} G_{small} & M_{distance} > l \\ G_{large} & M_{distance} <= l \end{cases}. \tag{22}$$

In the above criteria, l is a threshold, which can be defined according to the user requirements (in our experiment, we set the value of l as half of the largest gesture amplitude which can be performed by human body). We calculate the displacement of the gesture, when the displacement is larger than l, we define it as a large amplitude gesture G_{large}, otherwise we define it as a small amplitude gestures G_{small}.

In the example shown in Fig. 2a, the 3 wearable sensors A, B, C are put on the human wrist, elbow and shoulder, respectively. Since the lengths of the human upper arm and lower arm are both fixed, the distance between A and B, B and C are also fixed. In such example, we may refine the tracking error as follows. Let the length of the human upper arm is r_1 (the distance between A and B), and the human lower arm is r_2 (the distance between B and C). Since in this example, the shoulder is fixed, let its initial coordinate be (C_x, C_y, C_z). The calculated coordinate of point $A^{'}$ by improvement method 1 is ($A^{'}_x$, $A^{'}_y$, $A^{'}_z$). The calculated coordinate of point $B^{'}$ by improvement method 1 is ($B^{'}_x$, $B^{'}_y$, $B^{'}_z$). The final calculated coordinate $B_F(B_{Fx}, B_{Fy}, B_{Fz})$ for node $B^{'}$ is

$$\begin{aligned} B_{Fx} &= B^{'}_x + (r_1 * (B^{'}_x - C_x))/d_{B^{'}C} \\ B_{Fy} &= B^{'}_y + (r_1 * (B^{'}_y - C_y))/d_{B^{'}C} \\ B_{Fz} &= B^{'}_z + (r_1 * (B^{'}_z - C_z))/d_{B^{'}C} \\ d_{B^{'}C} &= \sqrt{(B^{'}_x - C_x)^2 + (B^{'}_y - C_y)^2 + (B^{'}_z - C_z)^2}. \end{aligned} \tag{23}$$

The final calculated coordinate $A_F(A_{Fx}, A_{Fy}, A_{Fz})$ for node $A^{'}$ is

$$\begin{aligned} A_{Fx} &= A^{'}_x + (r_2 * (A^{'}_x - B_{Fx}))/d_{A^{'}B^{'}} \\ A_{Fy} &= A^{'}_y + (r_2 * (A^{'}_y - B_{Fy}))/d_{A^{'}B^{'}} \\ A_{Fz} &= A^{'}_z + (r_2 * (A^{'}_z - B_{Fz}))/d_{A^{'}B^{'}} \\ d_{A^{'}B^{'}} &= \sqrt{(A^{'}_x - B_{Fx})^2 + (A^{'}_y - B_{Fy})^2 + (A^{'}_z - B_{Fz})^2}. \end{aligned} \tag{24}$$

In the above equation, $d_{B^{'}C}$ and $d_{A^{'}B^{'}}$ respectively denote the distance between point C and point $B^{'}$, and the distance between point $B^{'}$ and point $A^{'}$.

Since we can trace the trajectory of the sensors and the sensors are placed on the joints of the human body, combining with the correlation among the joints of the human body, we can precisely recognize the human gestures.

3.4 Recognition Algorithm

After obtaining the coordinates of each sensor placed on the human joints, we can recognize the human gesture by comprehensively considering the whole information. Suppose a person wears n sensors on the body, each sensor has displacement data along x, y and z axes, then each data of gesture is a $3 * n$ vector.

As shown in Fig. 3, assuming that D is our gesture library and there are a total of m gestures in D, then D is a $3 * n * m$ dimensional matrix, and we can use the following formula to identify gestures:

$$i^* = argmin||Ge - D_i||^2_2, i = 1, \ldots, n. \tag{25}$$

In the above formula, Ge represents the gesture vector to be recognized, and D_i represents the i gesture in the gesture library D. Finding an i^* which minimizes the formula, we can regard D_{i^*} as the gesture of Ge.

4 Experiment

In this section, we first introduce our implementation, followed by the investigation of key factors. At last, our motion tracking results in real time are given.

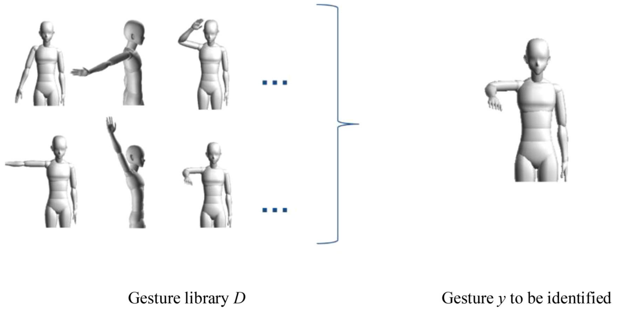


Fig. 3. Gesture library.

4.1 Implementation

In our experiment, we utilize the mpu6050 sensors. Each sensor contains a MEMS 3-axis accelerometer and a MEMS 3-axis gyro in a single chip. The low-pass filter algorithm is also integrated in the chips. The cutoff frequency is set as 44 Hz, since such setting will have both low latency and high filtering effect according to mpu6050 manual [58]. The communication module is nrf51822. Its core is ARM Cortex-M0. Keil is our programming and debugging tool. The part of the implementation code of the algorithm is in https://github.com/630440348/Fine-grained-and-Real-time-Gesture-Recognition.

Our framework chart is described in Fig. 4. We can divide the whole system into three parts: data preprocessing, displacement calculation and classification of gesture. In the first data preprocessing, we process the data collected from sensors and correct the hardware errors. In the following, we use a series of noise reduction method mentioned in the previous section to eliminate error. In particular, when data is missing, we also provide methods to supplement the missing value, as shown in Formulas (12), (13), and (14). In the second part, we use our recognition algorithms to calculate the displacement of the gesture movement mentioned in Section 3. At last in the part of classification of gesture, by using our gesture recognition algorithm, we are able to output the gesture and the gesture type.

Before running our algorithm, the sensors are put on the fixed positions of the human body. Each sensor will transmit its sensing data back to the server by using Bluetooth protocol [59]. The transmission time interval of each sensor is set as $10\ ms$. The recognition algorithm is run on the server. For each sensor, when the total number of received packets reaches 80, the location of such sensor will be calculated. Such procedure is repeated, and the trajectory of each sensor will be obtained.

4.2 Results of Refining Hardware Linear Distortion and Eliminating Gravity Impact

In this subsection, we randomly chose a sensor and test it in stationary state. Before running our algorithm, the obtained accelerometer data are shown in Fig. 5a. We see that, the acceleration data along the x and y axis are not equal to $0(m/s^2)$(due to the hardware linear distortion). Even worse, the acceleration data along the z axis are about $10(m/s^2)$(due to the impact of the gravity). After using our algorithm, the results are shown in Fig. 5b. We may see that, both the hardware linear distortion of each axis and the impact of gravity are successfully eliminated. The acceleration data of each axis are all close to $0(m/s^2)$ in stationary status.

4.3 Noise Reduction

In this subsection, we will investigate the impact of the five-spot triple smoothing method. As Shown in Fig. 6, we utilized the smoothing method to deal with the acceleration signal. The top sub figure is the original acceleration signal before processing. The bottom sub figure is the result after smoothing. According to the figure, we can see that, the five-spot triple smoothing method can reduce the high

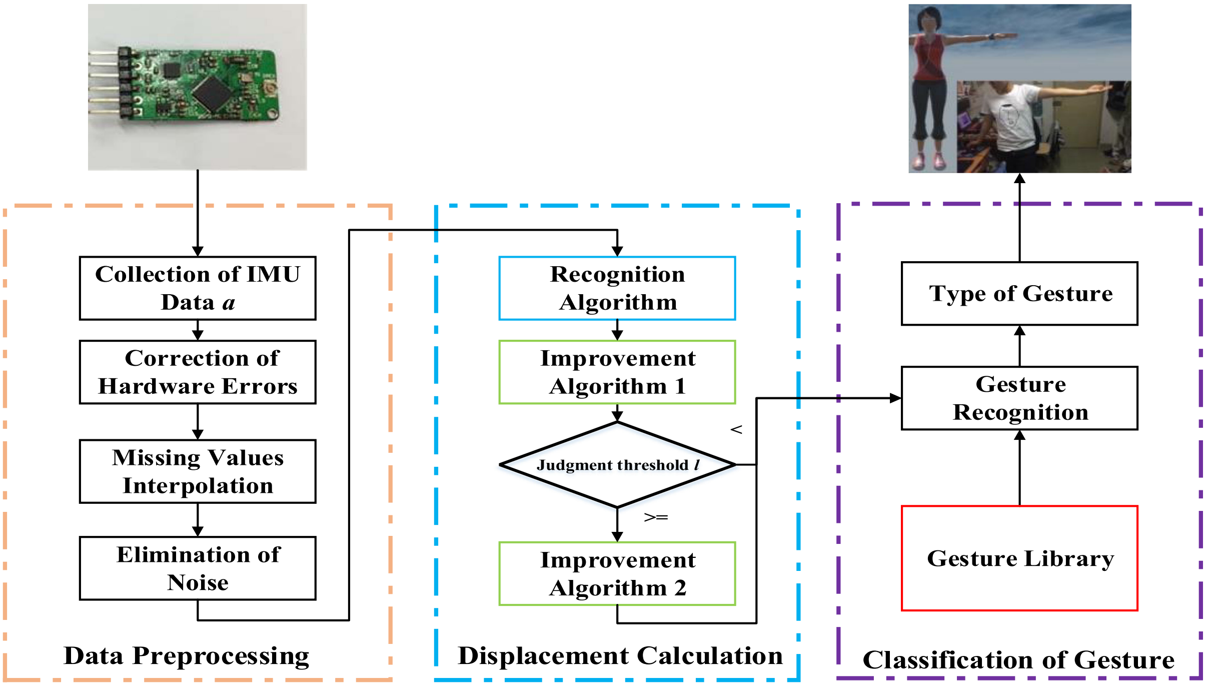
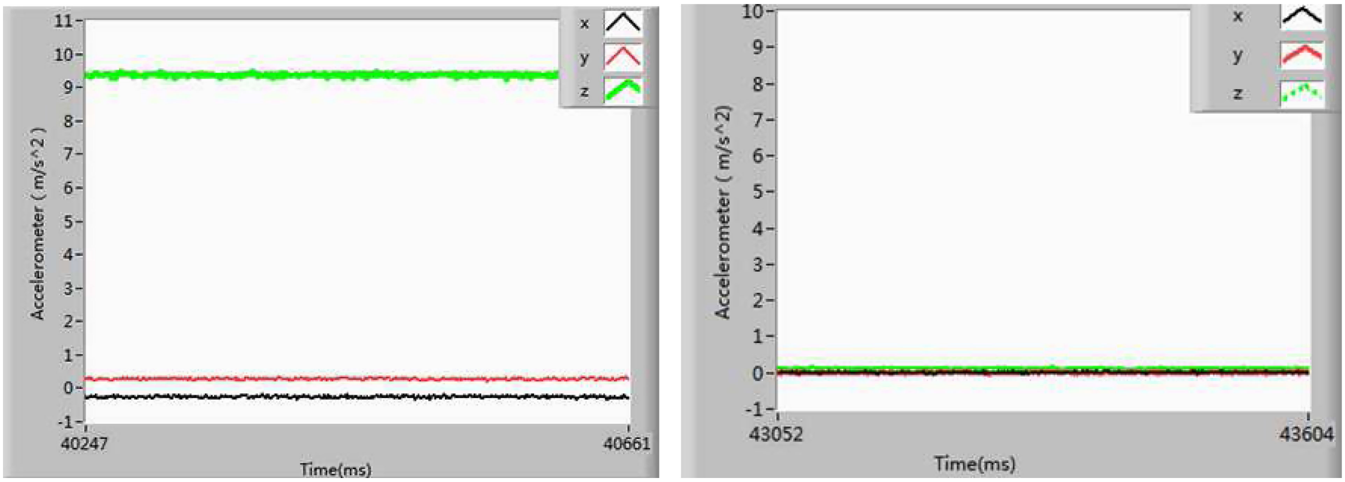


Fig. 4. Framework.



(a) Before refining hardware linear distortion and gravity elimination impact

(b) After refining hardware linear distortion and gravity elimination impact

Fig. 5. Data correction of gyroscope and accelerometer.

frequency random noise in the signal, which can help later displacement calculation.

4.4 Accuracy of the Gesture Recognition

In this subsection, we will test how our algorithm will be able to successfully track human gesture.

In our experiments, the sensors were placed on the arm. In total, we tested 40 sets of gestures by 10 people with different height and weight. The height of the target varies from $150\,cm$ to $182\,cm$, while the weight of the target varies from $45\,kg$ to $80\,kg$. The tested gestures are shown in Fig. 7. Our gestures include waving arms, lifting arms, dropping arms, etc. The moving range description of the tested gestures is shown in Table 2. We can see the minimum and maximum degree of the roll, pitch and yaw. The maximum and minimum displacement of the tested gestures are $10\,cm$ and $150\,cm$, respectively.

Fig. 8 shows the Cumulative Density Function (CDF) of accuracy, which is the difference between the calculated displacement and the real displacement. Our experiment results show that, our approach is able to successfully recognize the human gestures. We calculate the successful recognition rate as the number of successful recognition times over the number of whole tested gestures. How to identify the gesture is calculated by Equation (25). Experimental results show that the successful recognition rate can reach 100%. The experiment results are shown in Figs. 8a and 8d. We can see that, the average accuracy is about $0.07\,m$. By using the improvement methods 1 and 2, the average accuracy can reach about $0.06\,m$. Hence, the average accuracy can be improved by about 15%.

In order to investigate which improvement method contributes the most, we tested 20 rounds of gestures with large amplitude, and another 20 rounds of gestures with small amplitude, respectively. The experiment results are shown in Figs. 8b and 8e and Figs. 8c and 8f. We find that, for the gestures with large amplitude, the tracking accuracy is best when both the improvement methods are used. While for the gestures with small amplitude, improvement method 1 is the best. The reason for this is the following. For the improvement method 2, we use the correlation and limitation among

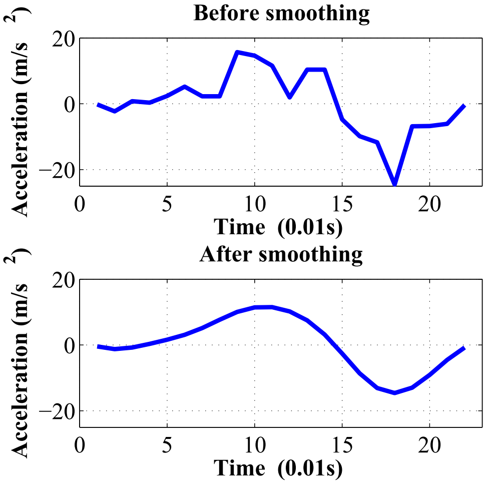


Fig. 6. Before and after smoothing.

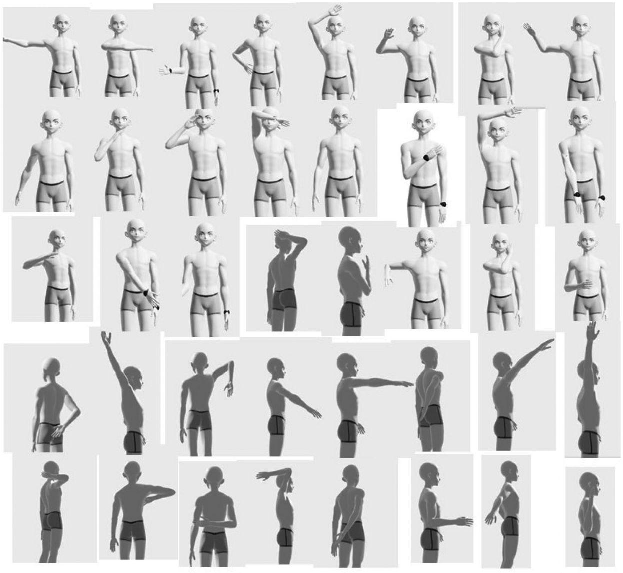


Fig. 7. The tested 40 gestures.

TABLE 2
Gesture Range

	Minimum value	Maximum value
Roll	-60°	180°
Pitch	-90°	145°
Yaw	-90°	145°
Displacement	10cm	150cm

body parts to avoid some abnormal calculated results beyond a certain limited range. However, the gestures with small amplitude usually will not go beyond this limitation. Therefore, in real application, we suggest using both the improvement methods. But in some scenario where only small human motions occur, we may just use the improvement method 1.

When more sensors are placed in the human joints or the sensors are placed on other parts of the body, we can recognize more complicate gestures.

4.5 Results of the Same Gestures With Different Amplitude

In this subsection, we will investigate how our algorithm can recognize the same gestures with different amplitude. We performed 20 rounds of test. In each round of test, we tested a certain gesture with two different amplitude. Each gesture is repeated twice. Fig. 9 is an example of such gesture. Fig. 9 shows the human gesture waving hand to the right. A, B and C are three wearable sensors. The solid line represents the real position of the human body, while the dashed line is the estimated position by our algorithm.

TABLE 3
Comparison of Different Gesture Amplitude

Gesture type	Xu's work [32]	Our system
Small amplitude	95.6%	100%
Both small and large amplitude	40%	100%

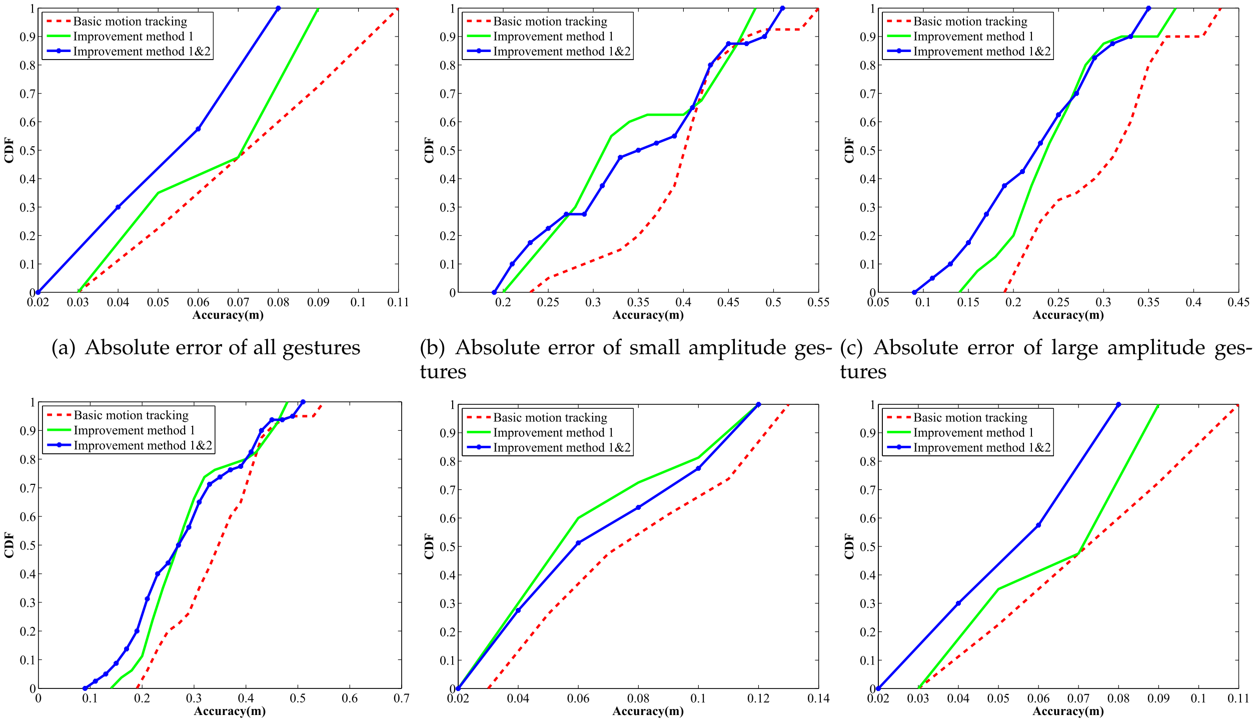
Our experiment results show that the successfully recognition rates of the gesture with large and small amplitude are both 100%. Therefore, our algorithm is able to successfully and comprehensively recognize the same gestures with different amplitude.

4.6 Comparisons With Other Algorithms

Our approach in gesture recognition is not limited by the predefined gestures. But in order to conveniently compare our algorithm with traditional algorithms, in this subsection, for the performance metric, we use the recognition rate instead of the previous performance metric accuracy.

The baselines we compared are listed below.

- Xu's work [32]. It is an automatic gesture segmentation algorithm to identify individual gestures in a sequence. It is able to extract a basic feature based on sign sequence of gesture acceleration. This method reduces hundreds of data values of a single gesture to a gesture code of 8 numbers. Finally, the gesture is recognized by comparing the gesture code with the stored templates.
- Cerqueira's work [60]. They used kinematic data and machine learning tools to recognize human upper body postures. The configuration that presented the



(a) Absolute error of all gestures

(b) Absolute error of small amplitude gestures

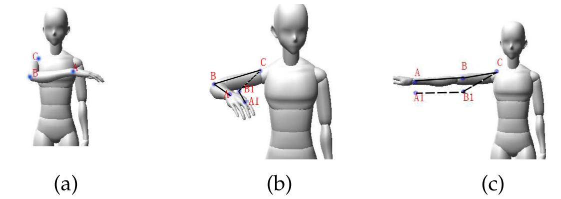
(c) Absolute error of large amplitude gestures

(d) Relative error of all gestures

(e) Relative error of small amplitude gestures

(f) Relative error of large amplitude gestures

Fig. 8. Accuracy of the motion tracking.



(a) (b) (c)

Fig. 9. The gesture is waving hand to the right. The solid line represents the real position of the human body, while the dashed line is the estimated position by our algorithm. (a) is initial position. (b) is the motion with small amplitude. (c) is the motion with large amplitude.

best results was a quadratic SVM classifier, with a Z-score normalization and the mRMR as dimensionality reduction algorithm.

- Kang's work [61]. They collected real time data from an IMU sensor attached to wrist and head of sport participants and used the decision tree-based classification scheme.
- Le's work [62]. It used a wearable device embedded with accelerometer and gyroscope sensors to recognize human hand gesture. It gave the best result when using Random Forests.
- Gochoo's work [52]. They employed a low-resolution infrared sensor-based wireless sensor network (WSN) and used a deep convolutional neural network (DCNN) to recognize yoga postures.
- Elforaici's work [63]. They leveraged CNN and SVM to recognize gestures using an RGB-D camera.
- Banos's work [64]. They presented an alternate approach based on transfer learning to opportunistically train new unseen or target sensor systems from sensor systems.
- Gibran's work [65]. They introduced a new benchmark dataset named IPN Hand with sufficient size, variety, and real-world elements. It utilized a 3D-CNN model in recognition.

Regarding to the tested gestures, we refer to the definition from Xu's work [32] and expanded to 40 different gestures including both large and small amplitude (referred to the definition in Equation (22)). The tested gestures are shown in Fig. 7. The detail setting is explained in Section 4.4. Experimental results show that, when recognizing the gestures with small amplitude, both our algorithm and Xu's algorithm can have high recognition rate. Our method outperformed Xu's algorithm by 4.4%. However, when taking both large and small amplitude gestures into account, the successful recognition rate of traditional Xu's work is reduced to 40%. It is noted that the successful recognition rate of our algorithm can still reach 100%, as Fig. 10 shows.

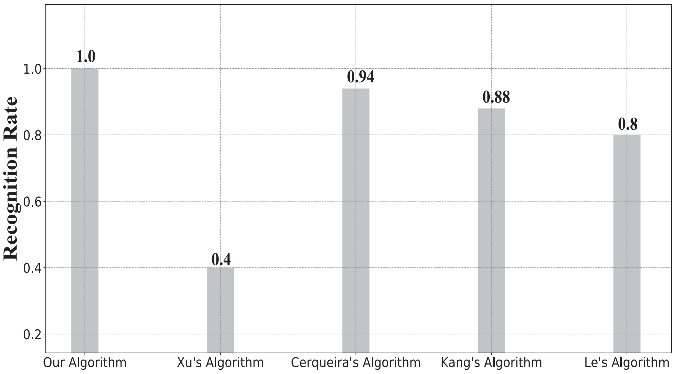


Fig. 10. Comparison with algorithms based on IMU data.

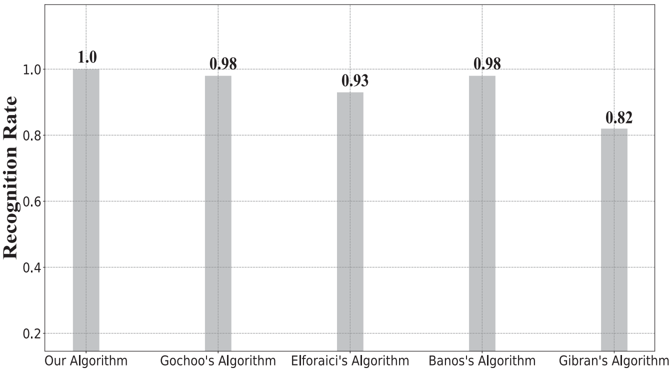


Fig. 11. Comparison with algorithms based on non-IMU data.

We further compare our algorithms with the baseline algorithms using IMU data. As shown in Fig. 10, our algorithm outperforms traditional Xu's algorithm, Cerqueira's algorithm [60], Kang's algorithm and Le's algorithm by $60\%, 6\%, 12\%$ and 20%, respectively.

On the other hand, we compare our algorithms with the baseline algorithms using non IMU data. As shown in Fig. 11, our algorithm outperforms Gochoo's work [52], Elforaici's work [63], Bano's algorithm, Gibran's algorithm by $2\%, 7\%, 2\%$ and 18%, respectively.

To sum up, although machine learning algorithms may have a high recognition rate, our algorithms still outperform them. Specifically, our algorithms do not require training in advance and are not limited by the predefined gestures, which can not be avoided by the machine learning algorithms.

4.7 System Latency

our system mainly contains the steps of data transmission, data processing and noise reductions. The latter two algorithms are implemented in our computer system with an Intel i5-3470K CPU and 8 GB RAM. Their running time can be negligible since it is too short and may vary with different computing systems. Therefore, the total latency is mainly determined by the transmission time, which depends on how much time for each sensor transmit its sensing data back to the server and how many data used in our tracking algorithm. In our setting, each sensor takes $10ms$ to transmit a packet. We will calculate its location after receiving 80 such packets. Therefore, the total latency is about $80 * 10 = 800ms = 0.8s$.

5 Conclusion

In this paper, we propose to exploit the sensing data from the wearable IMU devices to human body gesture recognition. Compared with traditional gesture recognition algorithms which require training, our approach does not have such requirement. We comprehensively study the sensing data, and propose various algorithms to refine the hardware linear distortion, eliminate the impact of the gravity, eliminate the hardware differences and effectively utilize the correlation and limitation among body parts to improve the tracking accuracy dramatically. As such, we can accurately recognize the human body gesture and track body motion in real time.

Experimental results have shown that our approach is able to successfully recognize the human gestures. The tracking accuracy of human body motion is about $0.06m$ and the recognition rate is 100%. Furthermore, our approach can recognize the same gestures with different amplitudes. Our algorithm is easily applied to recognize many complicated gestures, if more sensors are deployed on the other part of the body.

The future work is as follows. We will test different human body gestures in different environments. Moreover, more comprehensive algorithms to improve the tracking accuracy can be further investigated.

References

[1] L. E. Sucar, F. Orihuela-Espina, R. L. Velazquez, D. J. Reinkensmeyer, R. Leder, and J. Hern ández-Franco, "Gesture therapy: An upper limb virtual reality-based motor rehabilitation platform," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 3, pp. 634–643, May 2014.

[2] Y. Liu, Y. Yin, and S. Zhang, "Hand gesture recognition based on HU moments in interaction of virtual reality," in *Proc. 4th Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, 2012, pp. 145–148.

[3] Q. Wan, Y. Li, C. Li, and R. Pal, "Gesture recognition for smart home applications using portable radar sensors," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2014, pp. 6414–6417.

[4] K. Bouchard, A. Bouzouane, and B. Bouchard, "Gesture recognition in smart home using passive RFID technology," in *Proc. 7th Int. Conf. Pervasive Technol. Related Assistive Environ.*, 2014, pp. 1–8.

[5] J. Zeng, Y. Sun, and F. Wang, "A natural hand gesture system for intelligent human-computer interaction and medical assistance," in *Proc. 3rd Glob. Congr. Intell. Syst.*, 2012, pp. 382–385.

[6] Y. Zhang, J. Zhang, and Y. Luo, "A novel intelligent wheelchair control system based on hand gesture recognition," in *Proc. IEEE/ICME Int. Conf. Complex Med. Eng.*, 2011, pp. 334–339.

[7] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in *Proc. 21st IEEE Int. Symp. Robot Hum. Interactive Commun.*, 2012, pp. 411–417.

[8] A. Arbabian, S. Callender, S. Kang, M. Rangwala, and A. M. Niknejad, "A 94 GHz mm-wave-to-baseband pulsed-radar transceiver with applications in imaging and gesture recognition," *IEEE J. Solid-State Circuits*, vol. 48, no. 4, pp. 1055–1071, Apr. 2013.

[9] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial Wi-fi devices," *IEEE J. Selected Areas Commun.*, vol. 35, no. 5, pp. 1118–1131, May 2017.

[10] W. He, K. Wu, Y. Zou, and Z. Ming, "WiG: WiFi-based gesture recognition system," in *Proc. 24th Int. Conf. Comput. Commun. Netw.*, 2015, pp. 1–7.

[11] J. Yang, H. Zou, Y. Zhou, and L. Xie, "Learning gestures from Wifi: A Siamese recurrent convolutional architecture," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10763–10772, Dec. 2019.

[12] Y. Wu, K. Chen, and C. Fu, "Natural gesture modeling and recognition approach based on joint movements and arm orientations," *IEEE Sensors J.*, vol. 16, no. 21, pp. 7753–7761, Nov. 2016.

[13] S.-O. Shin, D. Kim, and Y.-H. Seo, "Controlling mobile robot using IMU and EMG sensor-based gesture recognition," in *Proc. 9th Int. Conf. Broadband Wirel. Comput., Commun. Appl.*, 2014, pp. 554–557.

[14] O. Sidek and M. A. Hadi, "Wireless gesture recognition system using MEMS accelerometer," in *Proc. Int. Symp. Technol. Manage. Emerg. Technol.*, 2014, pp. 444–447.

[15] H. Gunes and M. Piccardi, "Bi-modal emotion recognition from expressive face and body gestures," *J. Netw. Comput. Appl.*, vol. 30, no. 4, pp. 1334–1345, 2007.

[16] C.-F. Juang and P.-H. Wang, "An interval type-2 neural fuzzy classifier learned through soft margin minimization and its human posture classification application," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1474–1487, Oct. 2015.

[17] C.-H. Ling, C.-W. Lin, C.-W. Su, Y.-S. Chen, and H.-Y. M. Liao, "Virtual contour guided video object inpainting using posture mapping and retrieval," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 292–302, Apr. 2011.

[18] C.-F. Juang, C.-M. Chang, J.-R. Wu, and D. Lee, "Computer vision-based human body segmentation and posture estimation," *IEEE Trans. Syst., Man, Cybern.-Part A: Syst. Hum.*, vol. 39, no. 1, pp. 119–133, Jan. 2009.

[19] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2368–2377, Dec. 2014.

[20] D. Brulin, Y. Benezeth, and E. Courtial, "Posture recognition based on fuzzy logic for home monitoring of the elderly," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 5, pp. 974–982, Sep. 2012.

[21] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110–1120, Aug. 2013.

[22] Z. Ju, X. Ji, J. Li, and H. Liu, "An integrative framework of human hand gesture segmentation for human-robot interaction," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1326–1336, Sep. 2017.

[23] K. N. Krisandria, B. S. B. Dewantara, and D. Pramadihanto, "Hog-based hand gesture recognition using Kinect," in *Proc. Int. Electron. Symp.*, 2019, pp. 254–259.

[24] A. Dzikri and D. E. Kurniawan, "Hand gesture recognition for game 3D object using the leap motion controller with backpropagation method," in *Proc. Int. Conf. Appl. Eng.*, 2018, pp. 1–5.

[25] L. Shao, "Hand movement and gesture recognition using leap motion controller," Stanford Univ., Stanford, CA, USA, Rep. EE267, 2016.

[26] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1188–1192, Sep. 2016.

[27] J. Wang, L. Zhang, C. Wang, X. Ma, Q. Gao, and B. Lin, "Device-free human gesture recognition with generative adversarial networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7678–7688, Aug. 2020.

[28] S. H. Chae and S. Chung, "Blind interference alignment for a class of K-user line-of-sight interference channels," *IEEE Trans. Commun.*, vol. 60, no. 5, pp. 1177–1181, May 2012.

[29] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D tracking via body radio reflections," in *Proc. 11th USENIX Symp. Netw. Syst. Des. Implementation*, 2014, pp. 317–329.

[30] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," *Proc. ACM Conf. Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 1, 2021, pp. 1–27.

[31] Y. Zheng *et al.*, "Zero-effort cross-domain gesture recognition with Wi-Fi," in *Proc. 17th Annu. Int. Conf. Mobile Syst., Appl. Serv.*, 2019, pp. 313–325.

[32] 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.

[33] W. Zheng and D. Zhang, "Handbutton: Gesture recognition of transceiver-free object by using wireless networks," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 6640–6645.

[34] C. Zhang, C.-F. Lai, Y.-H. Lai, Z.-W. Wu, and H.-C. Chao, "An inferential real-time falling posture reconstruction for internet of healthcare things," *J. Netw. Comput. Appl.*, vol. 89, pp. 86–95, 2017.

[35] J. M. Allen, P. K. Asselin, and R. Foulds, "American sign language finger spelling recognition system," in *Proc. 29th Annu. Proc. Bioeng. Conf.*, 2003, pp. 285–286.

[36] S. P. Won, W. Melek, and F. Golnaraghi, "A fastened bolt tracking system for a hand-held tool using an inertial measurement unit and a triaxial magnetometer," in *Proc. 35th Annu. Conf. IEEE Ind. Electron.*, 2009, pp. 2703–2708.

[37] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, "A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors," *IEEE Sensors J.*, vol. 16, no. 16, pp. 6425–6432, Aug. 2016.

[38] X. Yun and E. R. Bachmann, "Design, implementation, and experimental results of a Quaternion-based Kalman filter for human body motion tracking," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1216–1227, Dec. 2006.

[39] H. Watanabe and T. Terada, "Improving ultrasound-based gesture recognition using a partially shielded single microphone," in *Proc. 2018 ACM Int. Symp. Wearable Comput.*, 2018, pp. 9–16.

[40] Y. Zhang, T. Gu, C. Luo, V. Kostakos, and A. Seneviratne, "FinDroidHR: Smartwatch gesture input with optical heartrate monitor," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, 2018, pp. 1–42.

[41] V. Becker, P. Oldrati, L. Barrios, and G. Soros, "Touchsense: Classifying finger touches and measuring their force with an electromyography armband," in *Proc. ACM Int. Symp. Wearable Comput.*, 2018, pp. 1–8.

[42] Y. Zhang, Y. Chen, H. Yu, X. Yang, and W. Lu, "Learning effective spatial–temporal features for SEMG armband-based gesture recognition," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6979–6992, Aug. 2020.

[43] T. K. Chan, Y. K. Yu, H. C. Kam, and K. H. Wong, "Robust hand gesture input using computer vision, inertial measurement unit (IMU) and flex sensors," in *Proc. IEEE Int. Conf. Mechatronics, Robot. Automat.*, 2018, pp. 95–99.

[44] Y. Zhang, Y. Yao, and Y. Luo, "An improved HMM/SVM dynamic hand gesture recognition algorithm," in *Proc. Adv. Display Technol. Micro/Nano Opt. Imag. Technol. Appl.*, 2015, pp. 74–80.

[45] S. Tan and J. Yang, "Fine-grained gesture recognition using WiFi," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2016, pp. 257–258.

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

[47] G. R. Naik, A. H. Al-Timemy, and H. T. Nguyen, "Transradial amputee gesture classification using an optimal number of SEMG sensors: An approach using ICA clustering," *IEEE Trans. Neural Rehabil. Eng.*, vol. 24, no. 8, pp. 837–846, Aug. 2016.

[48] C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," *IEEE Trans. Syst., Man Cybern.-Part A: Syst. Hum*, vol. 41, no. 3, pp. 569–573, May 2011.

[49] J. Zhao and R. S. Allison, "Real-time head gesture recognition on head-mounted displays using cascaded hidden Markov models," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2017, pp. 2361–2366.

[50] J. Wu, L. Sun, and R. Jafari, "A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors," *IEEE J. Biomed. Health Inform.*, vol. 20, no. 5, pp. 1281–1290, Sep. 2016.

[51] A. I. Bhuyan and T. C. Mallick, "Gyro-accelerometer based control of a robotic arm using microcontroller," in *Proc. 9th Int. Forum Strategic Technol.*, 2014, pp. 409–413.

[52] M. Gochoo *et al.*, "Novel IOT-based privacy-preserving yoga posture recognition system using low-resolution infrared sensors and deep learning," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7192–7200, Aug. 2019.

[53] K. Choi and H. Liu, "Low pass filter and band pass filter design," in *Problem-Based Learning in Communication Systems Using MATLAB and Simulink*. Hoboken, NJ, USA: Wiley-IEEE Press, 2016, pp. 55–65.

[54] What is the Difference Between Zero Offset and Zero Drift?. Accessed: Jun. 30, 2021. [Online]. Available: https://www.sensorsone.com/zero-offset-drift-difference/

[55] K. Shoemake, "Animating rotation with Quaternion curves," in *Proc. Conf. Comput. Graph. Interactive Techn.*, 1985, pp. 245–254.

[56] C. Croarkin and P. Tobias, *NIST/SEMATECH e-Handbook of Statistical Methods*. Gaithersburg, Md., USA: National Institute of Standards and Technology, 2012.

[57] D. H. Yu, *Natural Boundary Integral Method and Its Applications*. Beijing, China: Sci. Press, 2003.

[58] Accessed: Jun. 30, 2021. [Online]. Available: https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/

[59] B. A. Miller and C. Bisdikian, *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communication*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[60] S. M. Cerqueira, L. Moreira, L. Alpoim, A. Siva, and C. P. Santos, "An inertial data-based upper body posture recognition tool: A machine learning study approach," in *Proc. IEEE Int. Conf. Auton. Robot Syst. Competitions (ICARSC)*, 2020, pp. 4–9.

[61] M.-S. Kang, H.-W. Kang, C. Lee, and K. Moon, "The gesture recognition technology based on IMU sensor for personal active spinning," in *Proc. 20th Int. Conf. Adv. Commun. Technol.*, 2018, pp. 546–552.

[62] T.-H. Le, T.-H. Tran, and C. Pham, "The Internet-of-Things based hand gestures using wearable sensors for human machine interaction," in *Proc. Int. Conf. Multimedia Anal. Pattern Recognit.*, 2019, pp. 1–6.

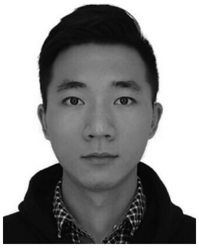
[63] M. E. Amine Elforaici, I. Chaaraoui, W. Bouachir, Y. Ouakrim, and N. Mezghani, "Posture recognition using an RGB-D camera: Exploring 3D body modeling and deep learning approaches," in *Proc. IEEE Life Sci. Conf.*, 2018, pp. 69–72.

[64] O. Banos *et al.*, "Opportunistic activity recognition in IoT sensor ecosystems via multimodal transfer learning," *Neural Process. Lett.*, vol. 53, pp. 3169–3197, 2021.

[65] G. Benitez-Garcia, J. Olivares-Mercado, G. Sanchez-Perez, and K. Yanai, "IPN hand: A video dataset and benchmark for real-time continuous hand gesture recognition," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 4340–4347.

Dian Zhang (Member, IEEE) received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010. She was a research assistant professor with the Fok Ying Tung Graduate School, HKUST. She was also an associate professor with Lingnan University, Hong Kong. She is currently an associate professor with Shenzhen university. Her research interests include big data analytics and mobile computing.

Zexiong Liao was born in 1996 in Shanwei City, China. He received the bachelor's degree in aerospace engineering from HuiZhou University in 2018 and the qualification for admission to the master's degree in engineering from Shenzhen University. His research interests include Internet of Things and big data analytics.



Wen Xie was born in 1996 in Jieyang City, China. He received the bachelor's degree in aerospace engineering from the Guangzhou college, South China University of Technology in 2018 and the qualification for admission to the master's degree in engineering from Shenzhen University. His research interests include Internet of Things and big data analytics.

Xiaofeng Wu received the master's degree in computer science and software engineering from Shenzhen University in 2018. Her master's thesis theme is fine-grained and real-time gesture recognition by Using IMU sensors. She is currently with the Shenzhen Research Institute of Big Data, China. Her research interests include Internet of Things, gesture recognition, and deep learning.

Haoran Xie (Senior Member, IEEE) received the PhD degree in computer science from the City University of Hong Kong. He is currently an associate professor with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong. He has authored or coauthored 247 research publications, including 119 journal articles. Among all 119 journal articles, there are 94 SCI/SSCI indexed and 13 SCOPUS indexed. His research interests include artificial intelligence, big data, and educational technology. He was the recipient of 14 research awards, including the Golden Medal and the Special Award from International Invention Innovation Competition in Canada, and five best paper awards from WI 2020, ICBL 2020, DASFAA 2017, ICBL 2016, and SECOP 2015. He is the editor-in-chief of the *Computers Education: Artificial Intelligence*, an associate editor for the *Array Journal*, *Australasian Journal of Educational Technology*, *Advances in Computational Intelligence*, and the *International Journal of Mobile Learning and Organisation*. He has successfully obtained more than 50 research grants, the total amount of which is more than HK$27 million. He is a senior member of ACM and a life member of AAAI.

Jiang Xiao received the BSc degree from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2009 and the PhD degree from Hong Kong University of Science and Technology in 2014. She is currently an associate professor with the School of Computer Science and Technology, HUST. Her research interests include blockchain, distributed computing, wireless indoor localization, and smart sensing. She has directed and participated in many research and development projects and grants from funding agencies such as the National Natural Science Foundation of China (NSFC), Hong Kong Research Grant Council (RGC), Hong Kong Innovation and Technology Commission (ITC), and industries like Huawei, Tencent, and Intel, and has been invited by NSFC in reviewing research projects. She was the recipient of several awards including the CCF-Intel Young Faculty Research Program 2017, Hubei Downlight Program 2018, ACM Wuhan Rising Star Award 2019, and the best paper awards from IEEE ICPADS/GLOBECOM/GPC.

Landu Jiang received the BEng degree in information security engineering from Shanghai Jiao Tong University, the MSc degree in computer science from the University of Nebraska-Lincoln, and the PhD degree with the School of Computer Science, McGill University. He is currently working toward the master's degree (minor) in construction management. His research interests include computer vision, machine learning, smart sensing, wearable and mobile computing, cyber-physical systems, green energy solutions, and online social networks.