An Air-writing Recognition Scheme employing Inertial Measurement Unit

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Submission Date: July 16th, 2021

Yuqi LUO

(5119FG38-2)

Advisor: Prof. Shigeru SHIMAMOTO

Research guidance: Research on Wireless Access Scheme

Acknowledgement

I would first like to thank my supervisor, Professor Shigeru Shimamoto, whose immense knowledge, and plentiful research experience have encouraged me in my academic research and daily life. I am extremely grateful for his invaluable advice, continuous support, and patience during my two-year master's study. Prof. Shimamoto supported me to find the research direction that I was really interested in and gave me much important advice about optimization. His outstanding personal charm and rigorous academic attitude will influence me all my life, urging me to become a rigorous and independent thinking researcher.

I would like to thank my research group leader A.P. LIU. She gave me much professional and invaluable guidance in my research and encouraged me to be a better person. Without her, my master’s career in the past two years would not be so smooth and positive. In terms of life, she often chats with me, which gives me a lot of courage and motivation to live alone in the foreign country.

I would also like to thank the lab member, my friend, Haosong Chen, who helped me collect a lot of data in my experiments and gave encouragement to each other. I also greatly thank many members of Shimamoto Lab, they care and help me a lot in the past two years.

Finally, I would like to express my heartfelt thanks to my beloved mother, Ms. Zheng. My achievements and personality are inseparable from her cultivation, and it is under her support that I can have these two years of valuable overseas study experience.

Time went by as the flowing river, now I will finish my master's study and bid farewell to my dear professors and classmates. The two years studying at Waseda University will be the most important experience in my life and will inspire me to become a better myself.

Abstract

Gesture recognition has been a popular research field under the trend of IoT and intelligent devices. Air-writing is the most challenging and crucial topic in the gesture recognition field. In this paper, we propose a wearable air-writing system that makes users can write the English alphabet in the three-dimensional space without any write rules. The proposed system is based on the Inertial Measurement Unit (IMU), and it uses dynamic time warping (DTW), as the main recognition algorithm. On this basis, we also use DTW and KNN (K-Nearest Neighbor) algorithm to make a creative combination, and a creative algorithm for time series analysis is proposed, Peak-number algorithm. In addition, to improve the recognition accuracy and take better advantage of the DTW algorithm, we present an adjustment system that gives some new optimization methods to the application of IMU and DTW.

In the experiment, the accuracy of recognition is 84.6% for the uppercase alphabet (from 'A' to 'Z') in user-dependent cases. And we also confirmed that the recognition method only based on the DTW algorithm is one kind of user-dependent method, which means this method is heavily dependent on personalization. For DTW enhancement algorithm, DTW-KNN algorithm that core idea is to use DTW distance to replace Euclidean distance in KNN, the accuracy can even reach 100% if the amount of input data is enough, but it also needs more computation and time. For our original algorithm, Peak-number algorithm, the accuracy of 26 letters can reach 75.3%. The most important advantage of the algorithm is that it almost does not need recognition time. For each letter, it only needs 0.0003s. The recognition of DTW and DTW-KNN usually takes several seconds or even more than ten seconds. By contrast, our proposed algorithm is very good in line with the requirements of real-time recognition, which is also needed in the actual application scene.

To sum up, in this paper a new wearable air-writing system, and a variety of recognition algorithms are studied and extended, and a creative time series recognition algorithm is proposed.

Keywords: Air-writing, DTW, DTW-KNN, Peak-number algorithm, gesture recognition, wearable devices, human-computer interface, optimization.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction
1.1 Background and Motivation

With the rapid development of computer science, electronic devices, and the Internet of Things around the world, human-computer interaction (HCI) schemes have become a very crucial part of intelligent devices application. HCI refers to the process of information exchange between users and systems to complete a certain task by using a certain conversational language and interacting in a certain way. Systems can be a variety of machines, such as computers, or they can be computerized systems and software. HCI function mainly depends on the input-output external equipment and the corresponding software to complete. Equipment for HCI mainly includes keyboard, display, mouse, various pattern recognition equipment, etc. With the development of computer technology, there are more and more operating commands and more and more powerful functions. With the development of pattern recognition, such as speech recognition, Chinese character recognition and other input devices, people have more efficient human-computer interaction methods besides keyboard input. In addition, in recent years the graphic human-computer interaction is also developing very fast. These human-computer interactions can be called intelligent human-computer interactions.

Gesture control is a new interactive method developed in recent years. Different from the general interactive methods such as keystrokes and voice, gesture control is easier to master and apply. Gesture control, as the name suggests, means that a human hand does not need to directly touch the machine, but makes corresponding postural changes in the air, and then gives the machine the input it needs to respond. Gesture controls offer several advantages over touch screens: for example, users can issue commands from a distance without touching the device. In the severe situation of the COVID-19 epidemic, the contactless gesture control HCI scheme is in line with the requirements of human health and epidemic prevention. Even after COVID-19 is controlled in the future, contactless HCI schemes will remain the trend of research and development. Gesture control is also an alternative to voice control, especially in public areas. For example, talking on a smart wearable device on the subway might be a little uncomfortable for some people and draw unwanted attention. Besides, gesture controls also expand the third dimension from a two-dimensional user interface. In previous research, gesture recognition has been wildly used in HCI applications such as sports sciences[1], intelligent wheelchair[2], gait detection[3], automatic television control systems [4], handwriting-based authentication systems [5], and air-writing.

Traditional gesture recognition provides a new modality for HCI. Some simple motion gestures are that users can memorize and apply easily. Besides, recognition algorithms of these simple gestures have high robustness and accuracy. However, for many complex usage scenarios, simple gestures have not enough information to achieve efficient and convenient HCI. Writing is the way and tool for human beings to record and express information with symbols. Until now, text input is still the most efficient way of human-computer interaction, because text contents include more information than voice or other input methods in unit input time. In general, text input mainly primarily through a keyboard, touch screen, or new speech-

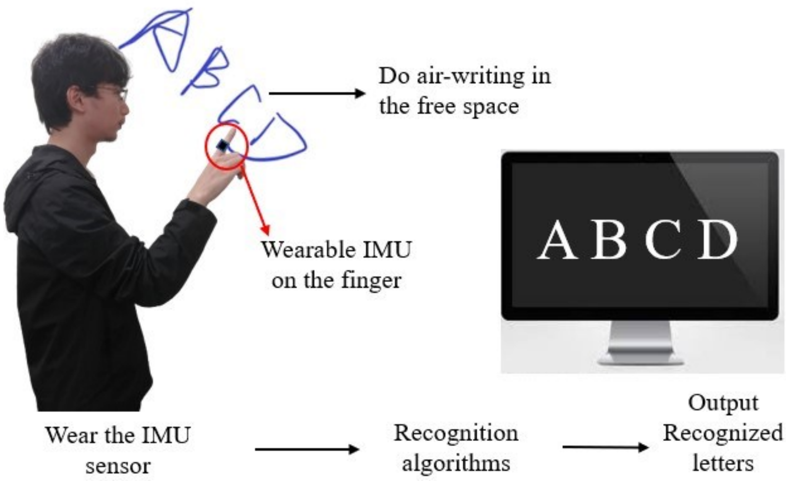


Fig. 1.1 The process of the proposed air-writing system.

to-text method. However, these old input methods may not meet the requirements of intelligent IoT scenarios. For example, in a smart home IoT system, we cannot equip every terminal or sensor with a touch screen or a keyboard. Air-writing is defined as writing linguistic characters or words in the free space by hand or finger movements [6]. It will be especially useful when users do not allow input text by traditional input methods. Besides, air-writing also has far-reaching application potential in the accessible field and wearable devices.

In this paper, we propose a wearable air-writing system that makes users can write the English alphabet in the three-dimensional space without any write rules. The recognition process of the proposed IMU is shown in Fig. 1.1.

1.2 Accessibility

Motor skills can be divided into fine motor skills and gross motor skills according to the muscles and range of movement involved. Fine motor skills are mainly achieved by small muscle movements (such as wrist and finger movements), carried out in a narrow space, requiring exquisite coordination of movement skills. Such as writing, typing, carving, embroidery, and other skills. Gross motor skills are the skills that use big muscles to achieve and require great strength and great movement. Such as running, swimming, playing ball games, weightlifting and so on. Undoubtedly, traditional writing is a fine motor skill because it requires subtle motions of the hand, the wrist, and fingers.

Fine motor disability is a person's inability to perform tasks that require a certain degree of dexterity, and it is a symptom, not the disease itself. The intact fine motor function involves complex coordination between numerous central and peripheral nervous system structures. The following neuroanatomical areas play crucial roles in fine motor control, and therefore any lesion can cause fine motor disability. Causes of lesions/damage include a space-occupying lesion, infection, stroke, toxins, autoimmune inflammation, metabolic, trauma, and congenital absence or abnormality. According to the data from NCBI (National Center for Biotechnology Information), the developmental disability in children in the US is 17.8%. There is a strong correlation between fine motor disability in children and development disorders. In adults, there are many diseases such as stroke (2.8% of the US population), rheumatoid arthritis (2% of North America), and traumatic brain injury (1.1% of the US population) will cause fine motor disability. In the medical field, Legibility and speed of writing have been used in children and adults as a way to determine patients' fine motor abilities. These data show that there are quite a few special people who cannot complete the traditional handwriting tasks very well. For some people with severe conditions, such as Parkinson's disease patients and Huntington's disease patients, they may not even be able to hold a pen.

The proposed wearable air-writing system can help the disabled with fine motor disabilities to complete writing tasks and carry out efficient human-computer interaction. The air-writing system will mainly play a role in two aspects. The first one is the air-writing system can transform fine movement into gross movement, and Fig. 1.2 shows this process in the proposed air-writing system. When the user wears the air-writing system and does air-writing, the written letters can be successfully recognized even the user only uses the arm without the

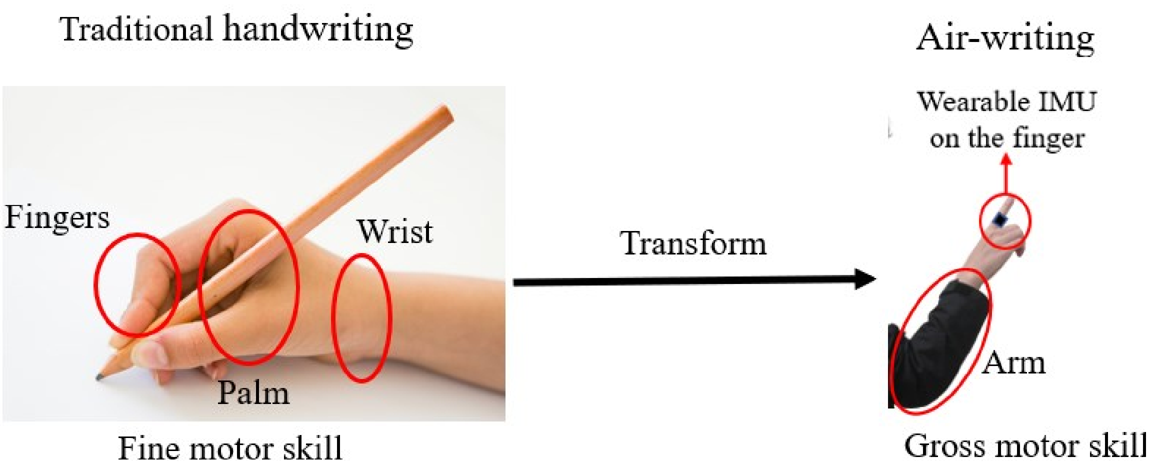


Fig. 1.2 The proposed system can transform fine motor movement into Gross motor movement

movements of fingers, the hand, and the wrist. The second one is the air-writing system is based on the DTW, DTW-KNN, Peak-Number-based algorithm. These algorithms are different from trajectory reconstruction or vision-based algorithms because they only focuses on the matching degree between the sample and the template, or some special recognition rules, not real letters. For example, even disabled users write some letters that are very different from the usual ones, the accuracy of air-writing will not decrease. Because personalization will ensure the sample and template match, thus ensuring a high recognition rate.

1.3 Related Works

Because of the rapid development of sensor and picture processing technologies, there are mainly two existing air-writing recognition schemes, IMU-based and vision-based. Because inertial sensors have some significant advantages, such as the small volume of hardware, low latency, and low computational cost, so they are wildly used in consumer electronic devices and the IoT field. In the initial research, researchers are aimed at combine pen with IMU. Wang et al. proposed an accelerometer-based digital pen. This digital pen can recognize handwritten digit from 0 to 9 and eight simple gestures, such as up, down, circle, and so on. Sepahvand et al. [8] introduced a new Persian/Arabic handwritten character recognition scheme with an IMU-based pen. It uses position signals collected by inertial pen to extract high-level geometrical features and then uses a new metric learning technique and a genetic programming algorithm. This scheme has high recognition accuracy of over 90%. However, even the sensor-based digital pen method has high accuracy results, it still does not meet the design concept of wearable devices. Zhou et al. [9] proposed an IMU-based method and mainly using trajectory estimation. This system can recognize ten-digits air-writing with 91.2% accuracy. In [10], Pan et al. proposed a scheme that uses built-in IMU in a smartphone to do air-writing. Then they investigate the performance of different letter recognition methods based on the reconstructed trajectories, including DTW and Hidden Markov Model (HMM)) and Convolutional Neural Network (CNN)). This method does not require additional sensors, and users can just hold their smartphones to finish air-writing. In [11], Tsai et al. proposed a vision-based air-writing method that mainly uses a camera to collect action information and a strong recognition algorithm. They present a novel reverse-time ordered stroke context, which can recognize the English alphabet and digital letters with 94.2% accuracy. In recent years, some creative and potential air-writing methods have been proposed. P. Wang et al. [12] proposed a new gesture air-writing tracking method based on 24 GHz SIMO Radar SoC. Their system has so many critical advantages such as very small size and enhanced sensitivity, without individual privacy risk and high robustness for environmental conditions, but the system can only recognize simple gestures and reconstruct the trajectory of ten-digits and English letters (without recognition).

1.4 Dissertation Outlines

The organization of the chapter is as follows:

In the chapter 1, the concept of HCI and the application of gesture control are introduced. In addition, this chapter also explains that under the general trend of various new technologies and demands, efficient air-writing technology is a research direction with great potential. Then the accessibility is explained, especially, it mainly expounds the causes of fine motor disorders, and its impact on the population is very large. The proposed air-writing system can convert fine motion into gross motion, so that patients can interact with the computer normally and efficiently by the proposed system. Finally, we summarize some related studies and results in the field of air-writing and elaborate their advantages and disadvantages.

The second chapter mainly expounds the hardware design of a wearable device. Firstly, we introduced the structure, function, and the choice of specific model of IMU. Then the details of the overall wearable device are introduced, including how it connects to the PC and the Pin of IMU and TTL-USB converter.

The third chapter mainly discusses three important algorithms being applied in the system, is the DTW algorithm, DTW - KNN algorithm, and Peak-Number algorithm respectively. We mainly discuss the mathematical principles of these three algorithms and their advantages in the application of air-writing systems. We also proposed some optimization tricks that make algorithms have better performance. Besides, we also discuss the time consumption of the algorithm, the most critical element of real-time recognition, in the practical application of air-writing system.

Besides, In chapter 4, several experiments and their results are described. For three different algorithms, we used the same data set to design different experiments and obtained the accuracy of comparative significance and other results.

In the final chapter, chapter 5, we summarize the conclusions from experiments that our proposed air-writing system has good performance and high accuracy. Future research plans are also described, focusing on wireless design, advanced algorithms, and integration with machine learning and deep learning.

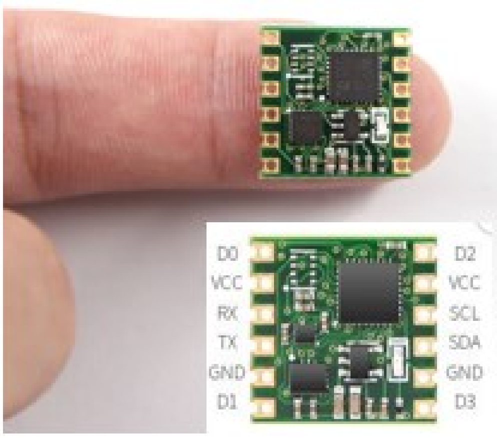
Chapter 2
Hardware Design
2.1 Inertial Measurement Unit

IMU is Inertial Measurement Unit, a kind of sensor used to detect and measure acceleration and rotational motion. Generally, an IMU includes a three-axis accelerometer and a three-axis gyroscope. The accelerometer can detect objects in the reference coordinate system independent triaxial acceleration signal, and the gyroscope can detect the angular velocity of movement. These information from the acceleration and the gyroscope can be used to calculate the movement trajectory and the posture of objects, so it has a very important application value in navigation and movement speculation. IMUs are mostly used in devices that require motion control, such as cars and robots. It is also used in the application of precision displacement calculation with attitude, such as inertial navigation equipment of submarine, aircraft, missile, and spacecraft, etc.

MEMS accelerometer is one of the earliest sensors in the MEMS field. After years of development, MEMS accelerometer design and processing technology have become increasingly mature. MEMS capacitive accelerometers are the cheapest, most common, and smallest of their kind. The operating principle boils down to changing the position of the anti-seismic mass suspended on the spring. One end of the spring is connected to the comb capacitor plate, while the other end is connected to the mass block. Under the action of the force acting on the sensor, the vibrating mass block moves, which causes the distance between the plate and the mass block to change, thus changing the capacitance. MEMS capacitive accelerometers are mainly used in wearable devices, mobile devices, and consumer electronics. One of the biggest advantages of MEMS accelerometers is that they can be mounted directly on the PCB. Disadvantages of MEMS systems include low measurement accuracy, especially in the case of higher amplitude and frequency measurements, which makes them unsuitable for specialized industrial applications.

Since the 1980s, more and more attention has been paid to the angular rate-sensitive MEMS gyroscopes. Its working principle is to use the conservation of angular momentum and the Coriolis effect to measure the angular velocity of moving objects. It is essentially a rotating object whose axis of rotation does not change with the rotation of the bracket carrying it. Like an accelerometer, the gyroscope's upper layer of moving metal forms a capacitor with the lower layer of metal. As the gyroscope rotates, the distance between it and the capacitor plate below will change, and the upper and lower capacitors will change accordingly. The change in capacitance is proportional to the angular velocity, from which we can measure the current angular velocity.

In our proposed air-writing system, we use an IMU which also includes a 3-axis magnetometer. The magnetometer uses Anisotropic magneto-resistance materials to detect the intensity of magnetic induction in space. The crystal structure of the alloy material is very sensitive to the external magnetic field, the strength of the magnetic field will lead to the change of AMR resistance value. In the proposed system, we use a WT901 (produced by Company WitMotion)



PIN	Function
➢ VCC	3.3-5V input supply
➢ RX	Serial data input, TTL interface
➢ TX	Serial data output, TTL interface
➢ GND	Ground
➢ DO	Analog input, digital input and output, PWM
➢ D1	Analog input, digital input and output, PWM, connect GPS
➢ D2	Analog input, digital input and output, PWM
➢ D3	Analog input, digital input and output, PWM
➢ SDA	I2C signal line
➢ SCL	I2C clock line

Fig. 2.1(a) The very small size IMU WT901 (b) Pin of IMU chip

which is based on InvenSense MPU9250. This IMU can achieve motion tracking function and output accelerated velocity, angle, angular velocity, and magnetic field intensity. Fig. 2.1(a)shows the very small size of the selected IMU, and Fig. 2.1(b) shows the Pin structure of this chip.

2.2 Wearable Device

In our proposed air-writing system, for data collection, the IMU is worn on the end of the index finger, like a ring. The realistic experiment picture is shown in Fig. 2.2. When the user moves the hand to do air-writing, the data produced by the IMU sensor will be transmitted to the PC by jumper wires and a TTL-USB conversion interface. The PC will start to conduct the recognition algorithm right after receiving the data. Fig. 2.3 shows the TTL-USB converter and its physical switch settings.

TTL-USB conversion interface is a six in one multifunctional serial port conversion module that can support the conversion of USB-TTL, USB-232, USB-485, TTL-485, 232-485, and

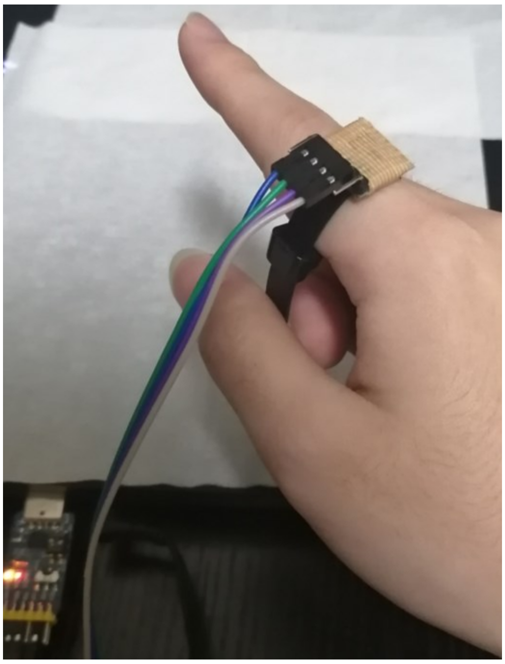


Fig. 2.2 The IMU on the end of the index finger

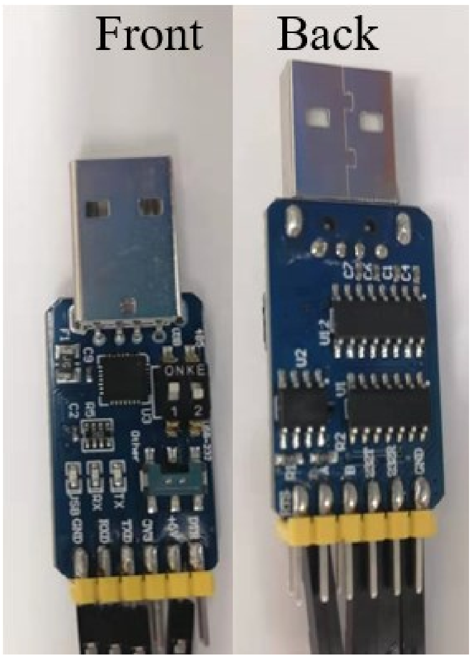


Fig. 2.3 The TTL-USB converter

TTL-232. It use USB 2.0 which is compatible with Windows XP/7/8/10 32bits/64bits, Linux, WinCE, Mac, Vista, etc. The indicator lamp uses red, yellow and green three colors to indicate the working state of the equipment. Com port selects the USB (yellow light) to be on normally, the data receives the RX (red light) flicker, the data sends the TX (green light) flicker. CP2102 from SILICON LAB is used as the main chip and the baud rate is 300 bps to 1.5 Mbps. Table 1 shows the Pin function of this converter.

Table. 1 The Pin function of converter

Name	Function
+5V	Power supply,5V input and output
3V3	Power supply,3.3V output
RX	Serial data input, TTL level
TX	Serial data output, TTL level
232R	Serial data input, 232 level
232T	Serial data output, 232 level
A	RS485 signal line A
B	RS485 signal line B
GND	GND
DTR	Data terminal preparation/ Control flow output
RTS	Request to send

Chapter 3
Methodology of Algorithms and Optimization
3.1 Dynamic Time Warping

In time series analysis, Dynamic Time Warp (DTW) is an algorithm to measure the similarity of two time series, and both series may have different time durations. Based on the idea of dynamic programming (DP), this algorithm solves the problem of non-point-to-point matching for different lengths of time series. DTW is a classical algorithm, in the 1970s and early 1980s, DTW was extensively used in speech recognition research [13]. Besides, DTW is efficient and competitive for small-scale samples without the training part. To sum up, the DTW algorithm can obtain the similarity of two time series, one as a sample and the other as a template. This process is shown in Fig. 3.1. And the smaller the similarity number, the higher the similarity between two time series, and vice versa.

In general, DTW is a method that calculates an optimal match between two given sequences

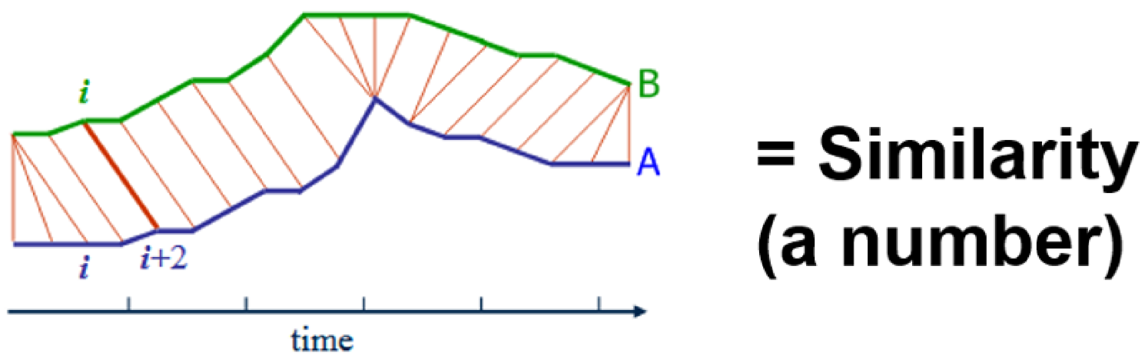


Fig. 3.1 Output of DTW

with certain restriction and rules:

1. Every index from the first sequence must be matched with one or more indices from the other sequence, and vice versa.

2. The first index from the first sequence must be matched with the first index from the other sequence (but it does not have to be its only match)
3. The last index from the first sequence must be matched with the last index from the other sequence (but it does not have to be its only match)
4. The mapping of the indices from the first sequence to indices from the other sequence must be monotonically increasing, and vice versa.

DTW is often compared with HMM (Hidden Markov Model) algorithm. Because DTW algorithm does not have an effective framework for training with statistical methods, and it is not easy to apply all kinds of knowledge at the lower level and the top level into the speech recognition algorithm, it is inferior to HMM algorithm in solving the problems of a large vocabulary, continuous speech and non-specific speech recognition. HMM is a kind of probabilistic model which is expressed by parameters and used to describe the statistical characteristics of the stochastic process. For isolated word recognition, HMM algorithm and DTW algorithm have little difference in recognition effect under the same condition. DTW algorithm itself is simple and effective, but HMM algorithm is much more complex. It needs to provide a large amount of speech data in the training stage, and the parameter model can be obtained through repeated calculation, while the DTW algorithm training almost does not need additional calculation.

Assuming there are two sequences that one is template sequence *T*, and another is sample sequence *S* as (1):

$$T = t_1, t_2, t_i, \dots, t_n$$
$$S = s_1, s_2, s_j \dots, s_m \tag{1}$$

f *n* is not equal to *m* in (1), the DTW can automatically match two sequences with different durations. If *n* is not equal to *m*, the time warping process can also achieve the best match between two sequences, and the best match will provide a more accurate similarity index than simply calculate the Euclidean distance of the template and the sample. Euclidean distance and the time warping are shown in Fig. 3.2.

For the matching operation, firstly we must construct a distance matrix M, and matrix element $M(i,j)$ represent the distance of $t_i$and s_j, it will be calculated by (2):

$$d(T_i, S_j) = (T_i - S_j)^2 \tag{2}$$

The second step of DTW is finding a path from the upper-left to the lower-right corner of the matrix that minimizes the sum of elements. The path will start from (1,1), and for a random point (i,j), it will have three predecessor candidates, i.e. $(i-1,j), (i,j-1), (i-1,j-1)$. Assuming the shortest distance of the optimal path is $L_{min}(i,j)$, so the recursive algorithm can be used to calculate the shortest path length as (3):

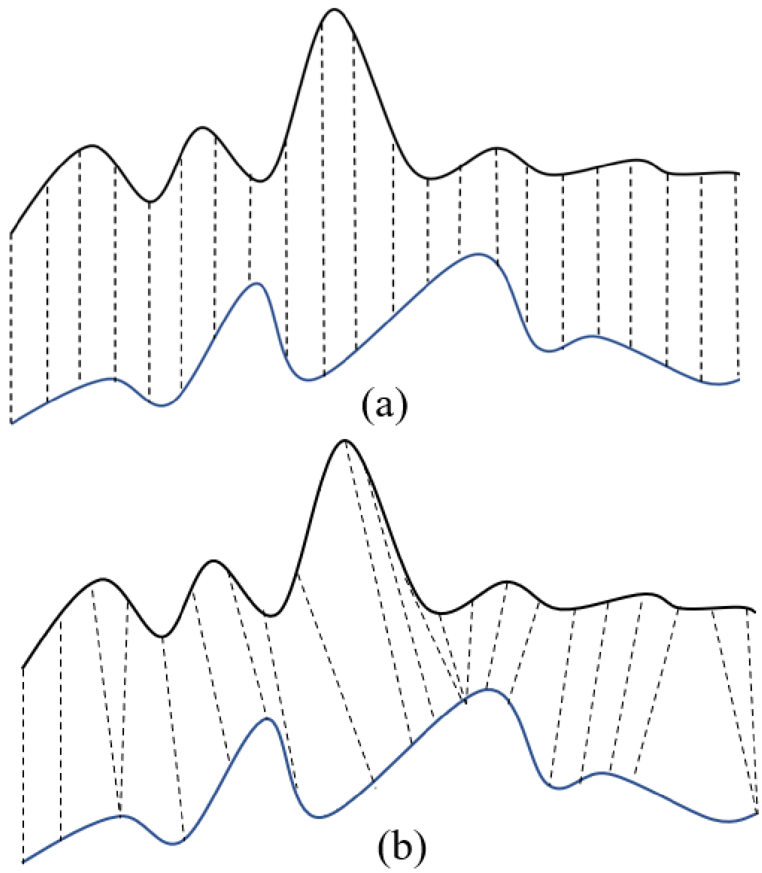


Fig. 3.2 Euclidean distance (a) and the Time Warping process (b) between two sequences.

$$L_{\min}(1,1) = M(1,1)$$

$$L_{min}(i,j) = M(i,j) + min\{L_{min}(i-1,j), L_{min}(i,j-1), L_{min}(i-1,j-1)\} \quad (3)$$

The time complexity and space complexity of DTW are both $O(M \cdot N)$. For improving the speed of the DTW algorithm, an optimized algorithm, FastDTW was proposed in [14]. FastDTW mainly uses three key operations: coarsening, projection and refinement. The time complexity of FastDTW is $N(8r + 14)$, and the space complexity is $N(4r + 7)$, where r is the radium parameter which controls the additional number of cells on each side of the projected path that will also be evaluated when refining the warp path. FastDTW significantly improves the speed of the algorithm with relatively high accuracy.

Fig. 3.3 shows the distance matrix *M* and a possible shortest distance path.

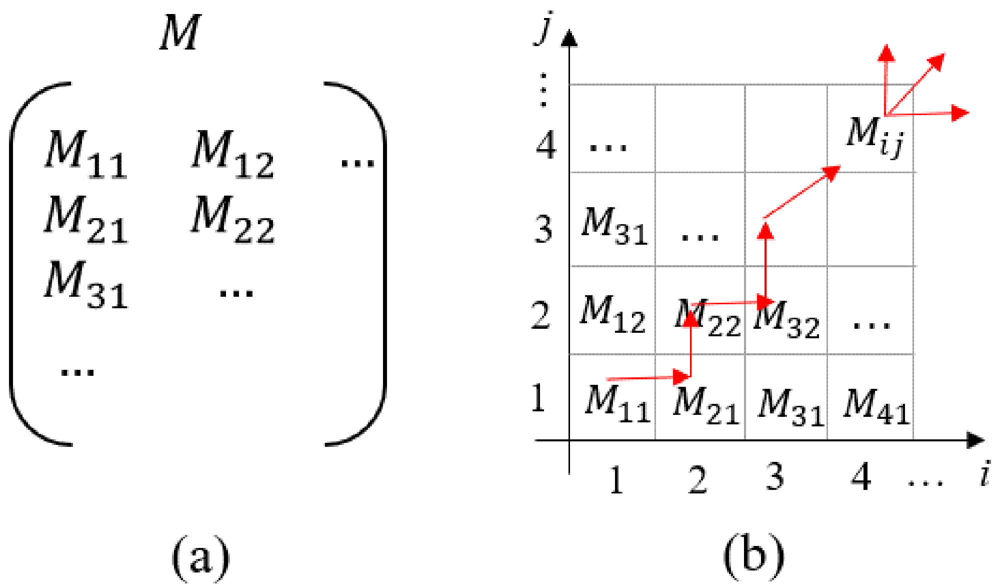


Fig. 3.3. The ditance matrix *M* (a). The red path in (b) means a possible shortest path to M_{ij}.

In chapter 2, we introduced the IMU which include a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. There is no doubt that the data output from the three sensors are all time series, based on the movement of the wearable device. The Fig. 3.4 shows the visualization of the time series of acceleration output under stationary state. Fig. 3.5 and Fig. 3.6 shows waveforms of gyroscope and magnetometer under stationary state. Fig. 3.7, Fig. 3.8, and Fig. 3.9 show the time series of 3-axis accelerometer, gyroscope and magnetometer when air-write the letter ‘A’. Fig. 3.10 shows a single axis, The time series of X-axis in Accelerometer when the user air-write letter ‘A’, letter ‘B’, letter ‘C’ and letter ‘D. From the time series figures, we can find that the IMU output time series is clear and reliable. In addition, the duration of writing action may be different according to the waveform of different letters, and the time duration of the same letter written at different times may also be different. For these characteristics, DTW as a recognition algorithm will be efficient and reliable, and fit the actual gesture recognition application scenarios.

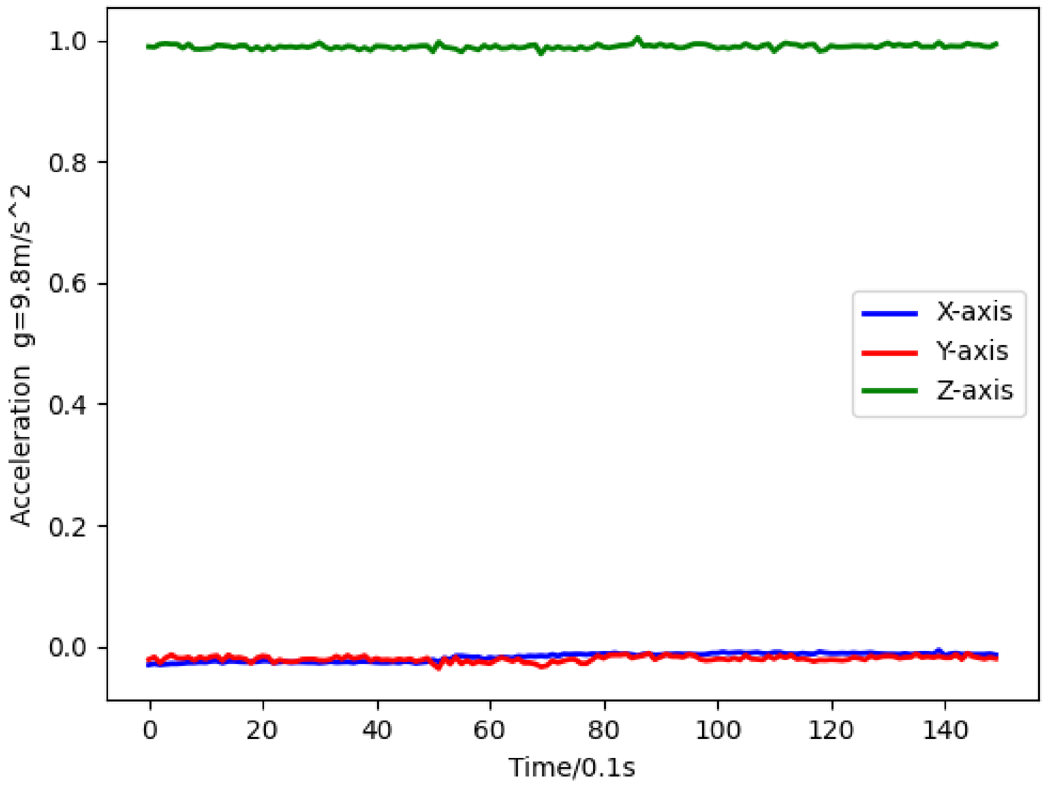


Fig. 3.4 The time series of 3-axis Accelerometer in stationary state

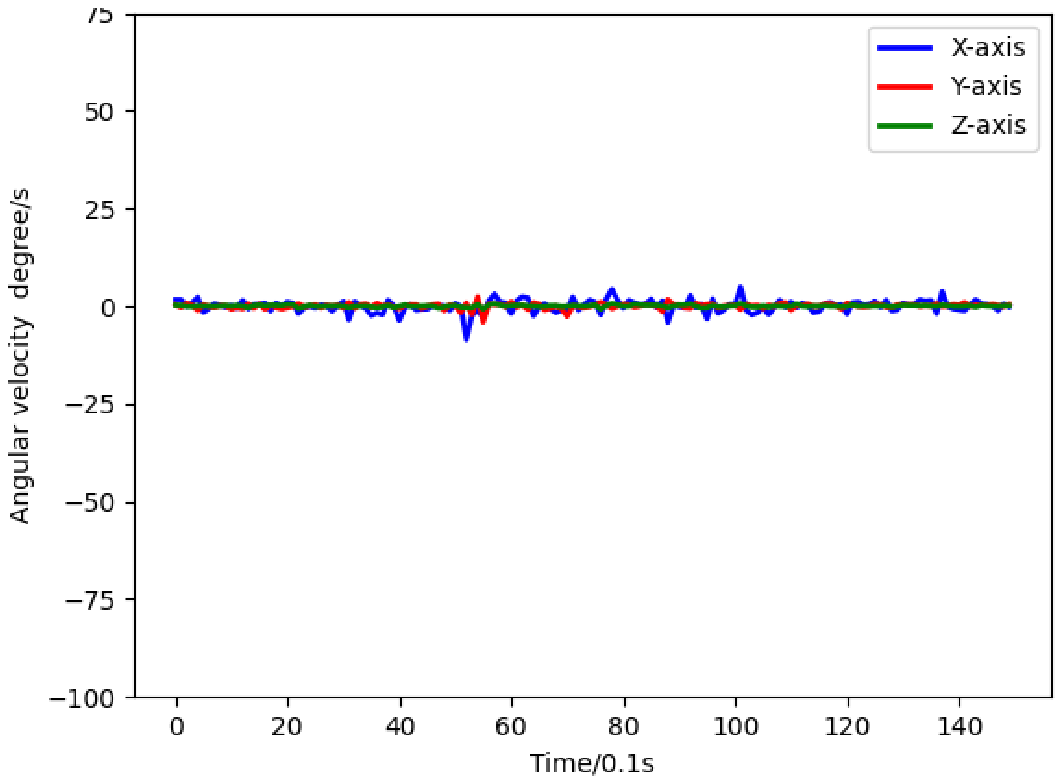


Fig. 3.5 The time series of 3-axis Gyroscope in stationary state

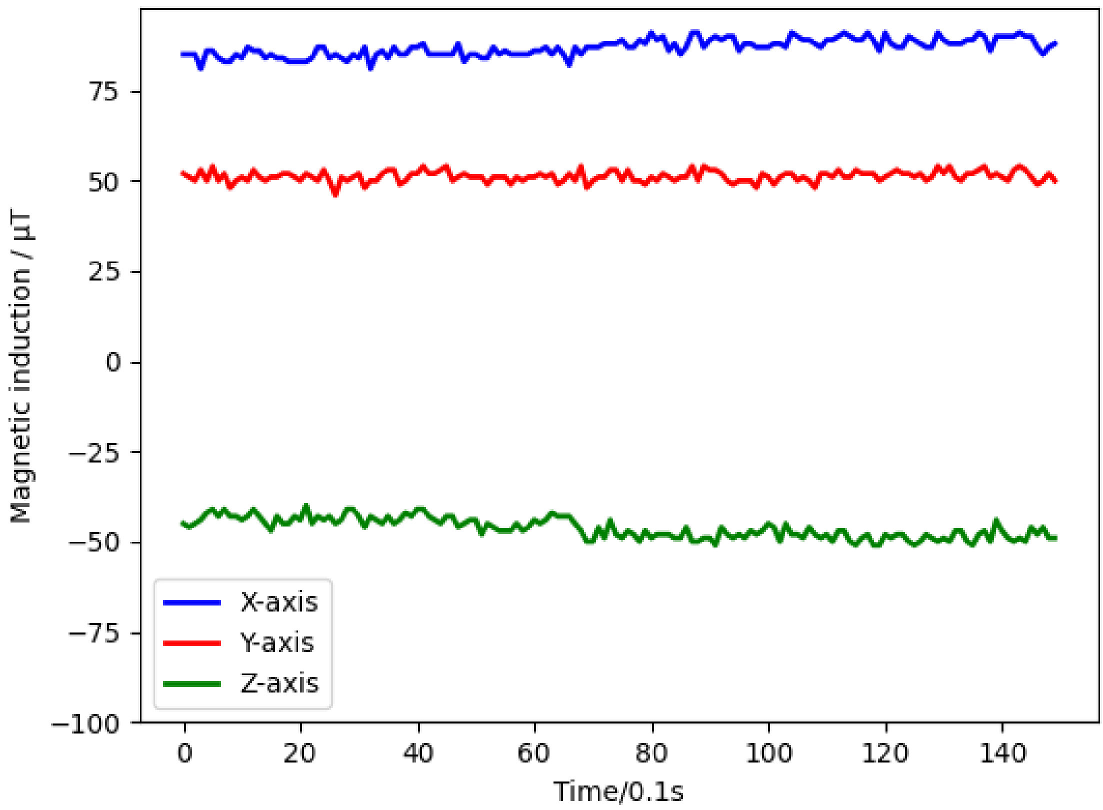


Fig. 3.6 The time series of 3-axis Magnetometer in stationary state

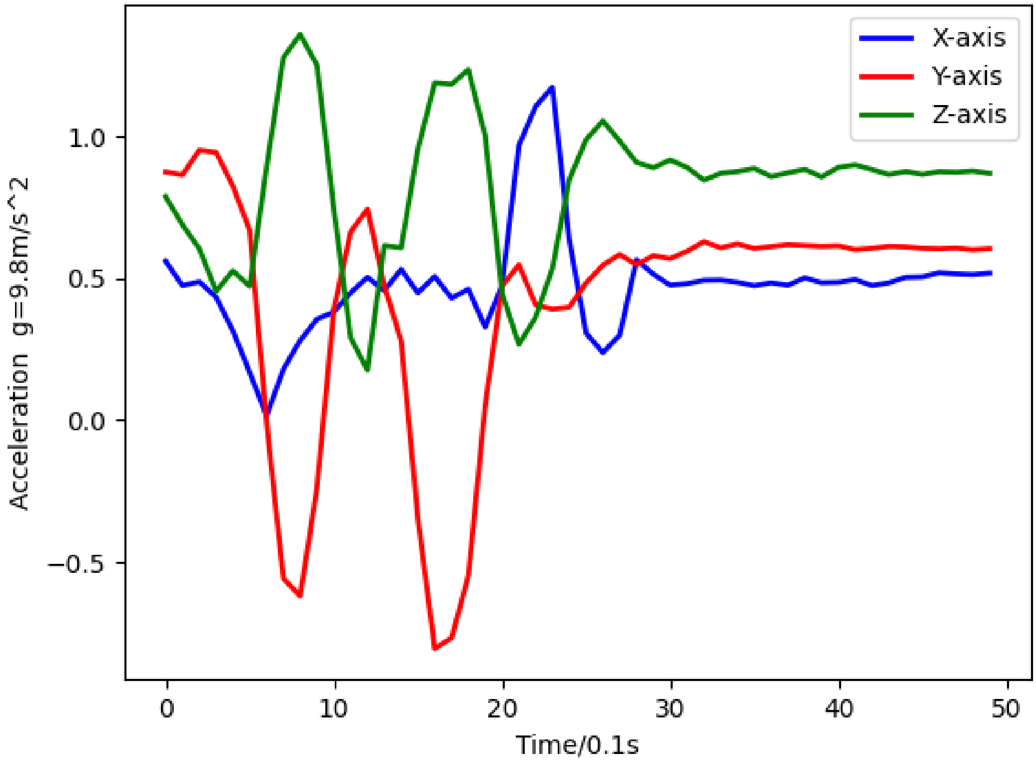


Fig. 3.7 The time series of 3-axis Accelerometer when air-write the letter ‘A’

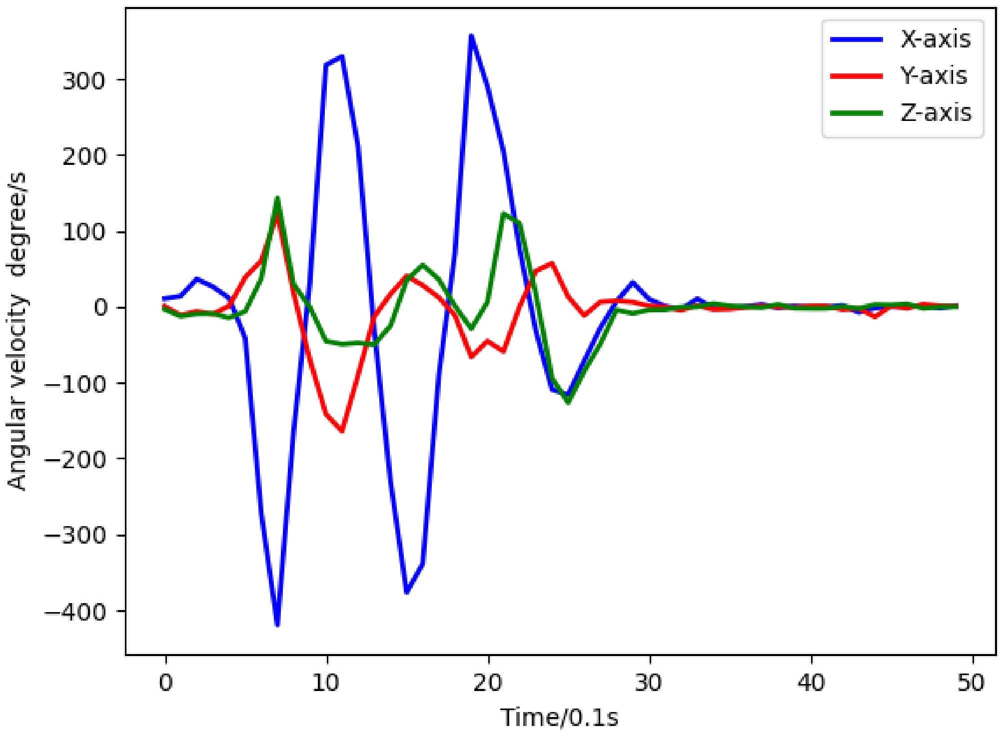


Fig. 3.8 The time series of 3-axis Gyroscope when air-write the letter ‘A’

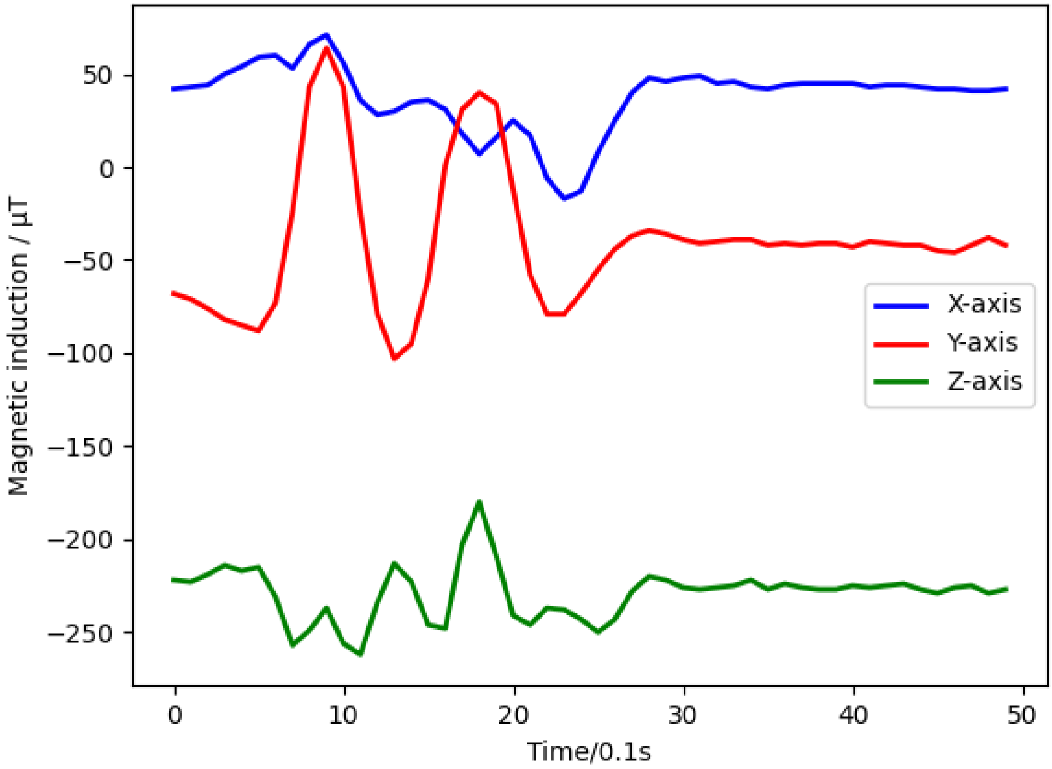


Fig. 3.9 The time series of 3-axis Magnetometer when air-write the letter ‘A’

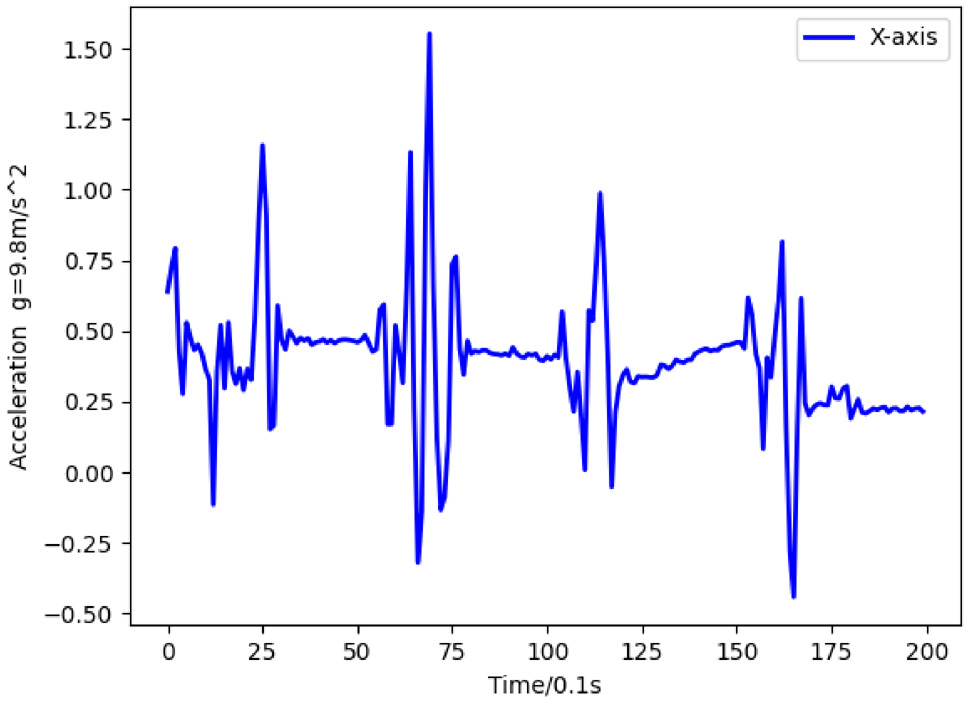


Fig. 3.10 The time series of X-axis in Accelerometer when the user air-write letter ‘A’, letter ‘B’, letter ‘C’ and letter ‘D’

3.2 K-nearest Neighbors and DTW-KNN

KNN (K-Nearest Neighbor) method, originally developed by Evelyn Fix and Joseph Hodges in 1951 [16]. And then expended by Cover. T and Hart. P in 1968 [17]. It is a relatively mature method in theory and one of the simplest machine learning algorithms. The idea of this method is very simple and intuitive: if most of the K most similar (that is, the closest in the feature space) samples of a sample belongs to a certain category, then the sample also belongs to this category. In the classification decision, the method only determines the category of the samples is divided according to the category of the nearest one or several samples.

In general, the KNN classification algorithm includes the following six steps:

1. Prepare data and preprocess the data.
2. Calculate the distance between the point in the known category data set and the sample point.
3. Sort by increasing distance.
4. Select K points with the smallest distance from the sample point.
5. Count the occurrence frequency of the category of the first K points.
6. The category with the highest occurrence frequency of the first k points is returned as the prediction classification of the sample point.

Fig. 3. 11 shows the main idea of KNN.

The Minkowski distance or Minkowski metric is a measure in Euclidean space, viewed as a generalization of Euclidean distance and Manhattan distance.

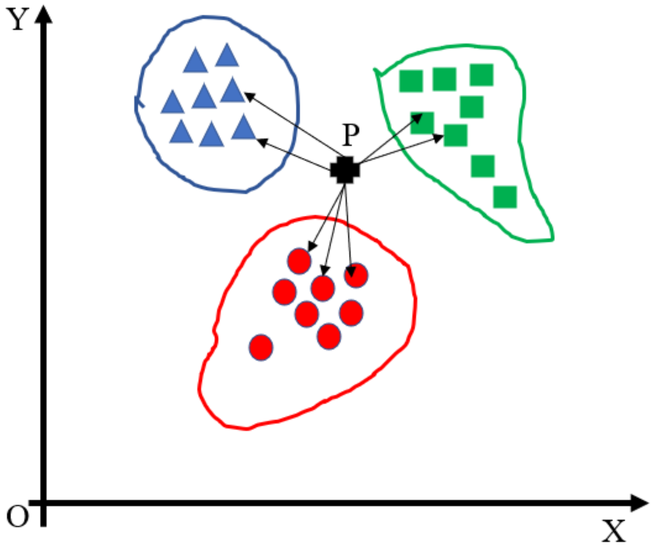


Fig. 3.11 The concept of KNN algorithm

For two points:

$$X = (x_1, x_2, x_3, \dots, x_n)$$

$$Y = (y_1, y_2, y_3, \dots, y_n) \tag{4}$$

The Minkowski distance is defined as:

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{5}$$

The Manhattan distance comes from the city block distance, which is the result of summing up the distance in multiple dimensions. When p is 1, the distance will be Manhattan distance:

$$dist(X,Y) = \sum_{i=1}^{n} |x_i - y_i| \tag{6}$$

Euclidean distance is the most common measure of distance, which measures the absolute distance between points in multidimensional space. When p is 2, the distance will be Euclidean distance:

$$dist(X,Y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{7}$$

Because the calculation is based on the absolute value of the characteristics of each dimension, the Euclidean measurement needs to ensure that the indicators of each dimension are at the same scale level. For example, the use of Euclidean distance for two indicators with different units of height (cm) and weight (kg) may invalidate the results.

Chebyshev distance comes from the moves of the king in chess. Extending to multidimensional space, Chebyshev distance is the Minkowski distance as p approaches infinity:

$$dist(X, Y) = \lim_{p \to \infty} (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}} = max|x_i - y_i| \tag{8}$$

fact, the Manhattan distance, Euclidean distance, and Chebyshev distance above are all applications of Minkowski distance under special conditions. In the mentioned step 2, we usually calculate the Euclidean distance for KNN algorithm. Fig. 3.12 shows the different K value choices. When K is 3, the classification result is Class 2, but when K is 7, the classification result will be Class 1.

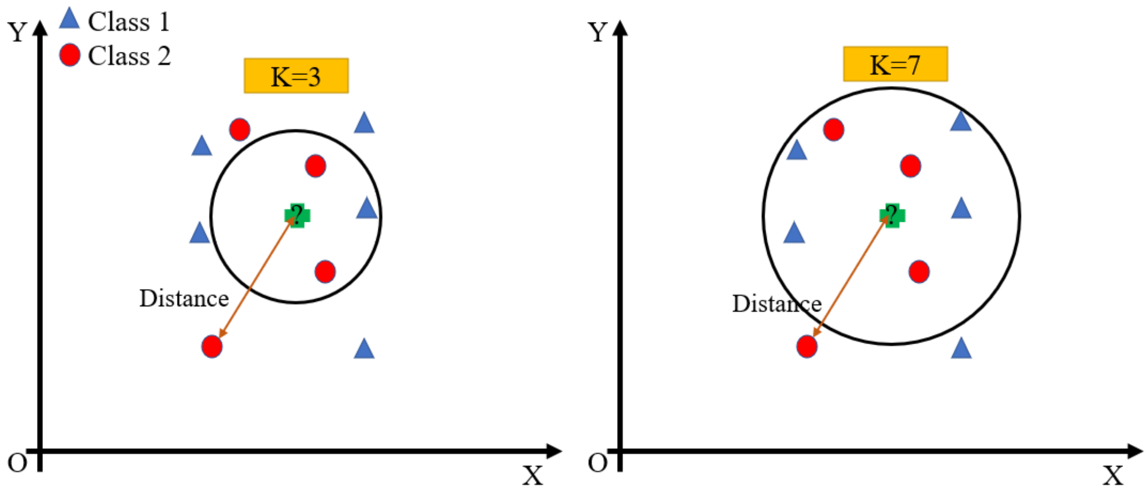


Fig. 3.12 Different K value choices will affect the results of the same classification problem

If a smaller value of K is selected, it is equivalent to using the training instance in the small neighborhood to make the prediction, and the approximation error of learning will decrease, and only the training instance that is close to the input instance will play a role in the prediction result. The single disadvantage is that the estimation error of learning will increase, and the prediction result will be sensitive to the instance point division in the nearby neighbor. If the adjacent instance point happens to be noise, the prediction will be wrong. In other words, the decrease of K value means that the overall model becomes more complex, and the division is not clear, so overfitting is likely to occur.

If a larger value of K is selected, it is equivalent to using training examples in a larger neighborhood to make predictions. Its advantage is that the estimation error of learning can be reduced, but the approximate error will increase, that is, the prediction of input examples is not accurate. If K is worth increasing, it means that the overall model becomes simpler. (Approximate error: can be understood as the training error of the existing training set. Estimation error: Can be understood as a test error on a test set.)

KNN method is simple, easy to understand, easy to implement, no need to estimate parameters. A major deficiency of this algorithm in classification is that when the samples are unbalanced, such as the sample size of one class is large while the sample size of other classes is small, it may lead to that when a new sample is input, the samples of the large-size class in the K neighbors of the sample are in the majority. Another disadvantage of this method is that it requires a large amount of computation, because the distance between each text to be classified and all known samples must be calculated to obtain its K nearest neighbors.

From the introduction and output graph of IMU in Section 1 of Chapter 3, we can know that IMU mainly outputs multiple time series. However, the KNN algorithm uses the distance between the sample points to be classified and each point in the data set as the main input (It is also possible to enter information about the location of data sets and sample points). When we want to use KNN to analyze time series, the most critical part is how to calculate the distance between two time series. In the introduction part of the DTW algorithm, we profoundly expounded the advantages of DTW for the analysis of two time series, and it can get a number representing the similarity, or the distance, of two time series. So, the main idea is to use the distance of two time series obtained by DTW algorithm to replace the Euclidean distance in the second step of KNN algorithm. In [15], the author proposed a method that DTW is applied for speech feature matching and KNN is employed as a classifier, this work is like the proposed DTW-KNN algorithm and inspires our work.

The algorithm is essentially a multiple iteration of the KNN-based DTW algorithm, so it will undoubtedly perform better than the classical DTW algorithm. But this superior performance comes at a price. The DTW-KNN algorithm is the brute force algorithm, which is essentially the calculation of DTW distance for a large number of time series. Therefore, it is difficult for this algorithm to perform well in large scale or large data sets, and the main disadvantage is time consumption. The time consumption in the experiment will be explained and shared in detail in the experiment chapter 4.

3.3 Peak-number Algorithm

The DTW and DTW-KNN algorithms described above are all for the analysis of the whole air-writing movement data. The advantage of these algorithms is that they don't miss any information in a complete air-writing movement. However, its drawbacks are also obvious that not all movement information is necessary for a successful and efficient recognition, besides, the computations that include all the information require strong computing power support and relatively long recognition time. For these inherent characteristics of air-writing recognition, Peak-number algorithm, a new algorithm is proposed in this paper. Obviously, pattern recognition itself is related to the classification problem, and the classification problem is the most critical similarity. In other words, if similarities and differences between samples and templates can be found, a successful classification task can be performed. The main idea of Peak-number algorithm is that different air-writing letters have different numbers of upward peak points and downward peak points. This kind of 'difference' will ensure that we can successfully recognize among different letters of the English alphabet. Fig. 3.13 shows this

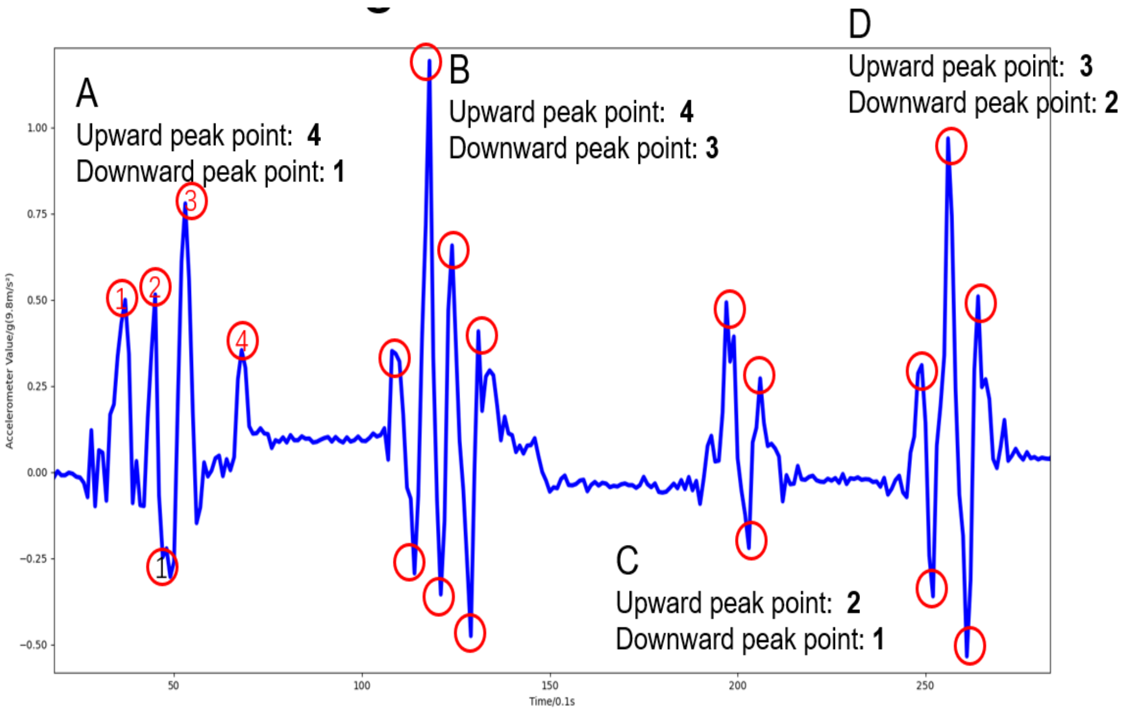


Fig. 3.13 Different air-writing letters have different peak points information

kind of 'difference' in the X-axis of the accelerometer, IMU.

In general, the Peak-number algorithm includes the following four steps:

1. Prepare data and preprocess the data which collected by IMU.
2. Calculate the number of upward peak points and downward peak points for a sample.
3. Compare the result of step 2 with standard peak point numbers for each letter.
4. The sample will be recognized as the matched standard letter.

The effect (X-axis of accelerometer) of the automatic peak finding system is shown in Fig. 3.14. It is designed in Python 3.8. It uses the same data as in Figure 3.13, and by comparison, it can be found that the results of automatic peak finding are completely correct. In [18], many peak searching algorithms and applications are clearly explained, in general, automatic peak searching is a relatively fast and simple task

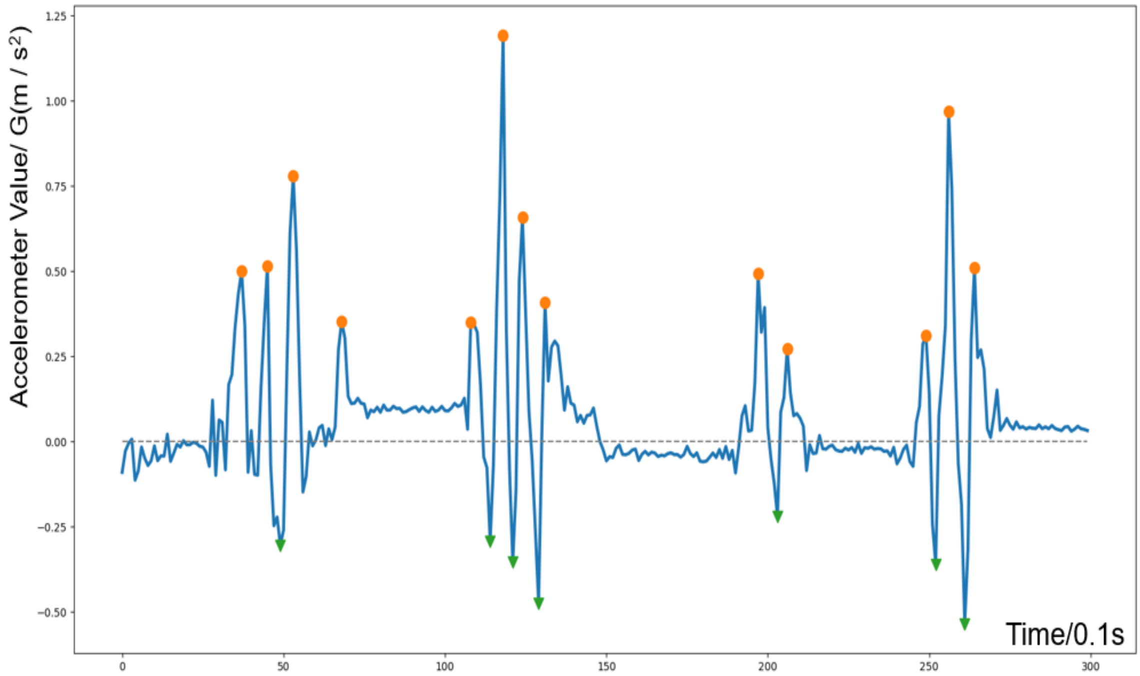


Fig. 3.14 The result of automatic peak finding for air-writing letters, ‘A’, ‘B’, ‘C’, ‘D’.

As we have mentioned, the IMU we have chosen has nine axes. However, in the actual experiment, we find that not all axes are suitable for using the Peak-number algorithm we proposed. Because on many axes, the peak information of the data is very unclear and messy, and that leads to if we use the data on these axes to extract information of peaks, the accuracy of the air-writing system will be very low. Fig. 3.15 shows the waveform that is not good enough from accelerometer X-axis and magnetometer X-axis. Unfortunately, not only X-axis, all the axes from the accelerometer and magnetometer are not perfect enough to make them difficult to use in the Peak-number algorithm.

On the contrary, the data waveform of the three axes of the gyroscope is very clear and perfect, which is very suitable for the Peak-number algorithm we proposed. Fig. 3.16 shows the waveform of gyroscope X-axis. Therefore, in the actual experiment, we mainly use the data from the three axes of the gyroscope to carry out air-writing recognition.

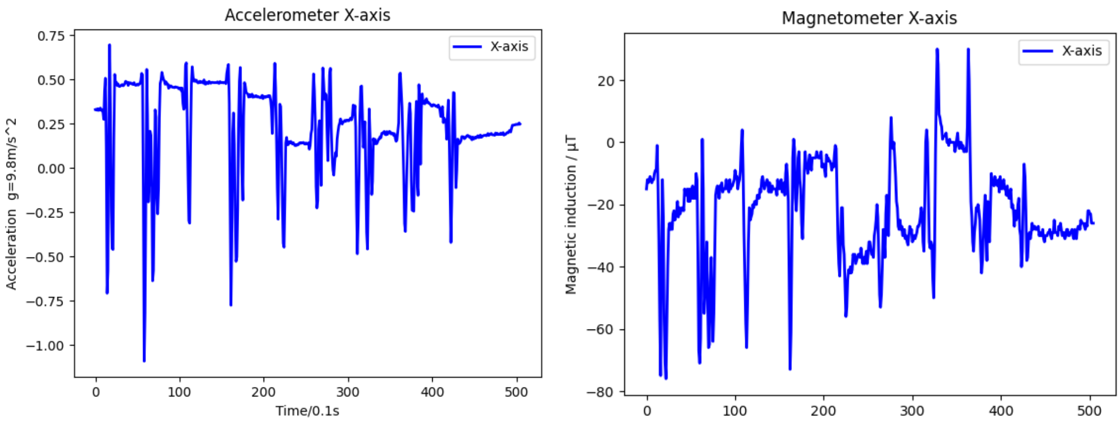


Fig. 3.15 bad performance axes

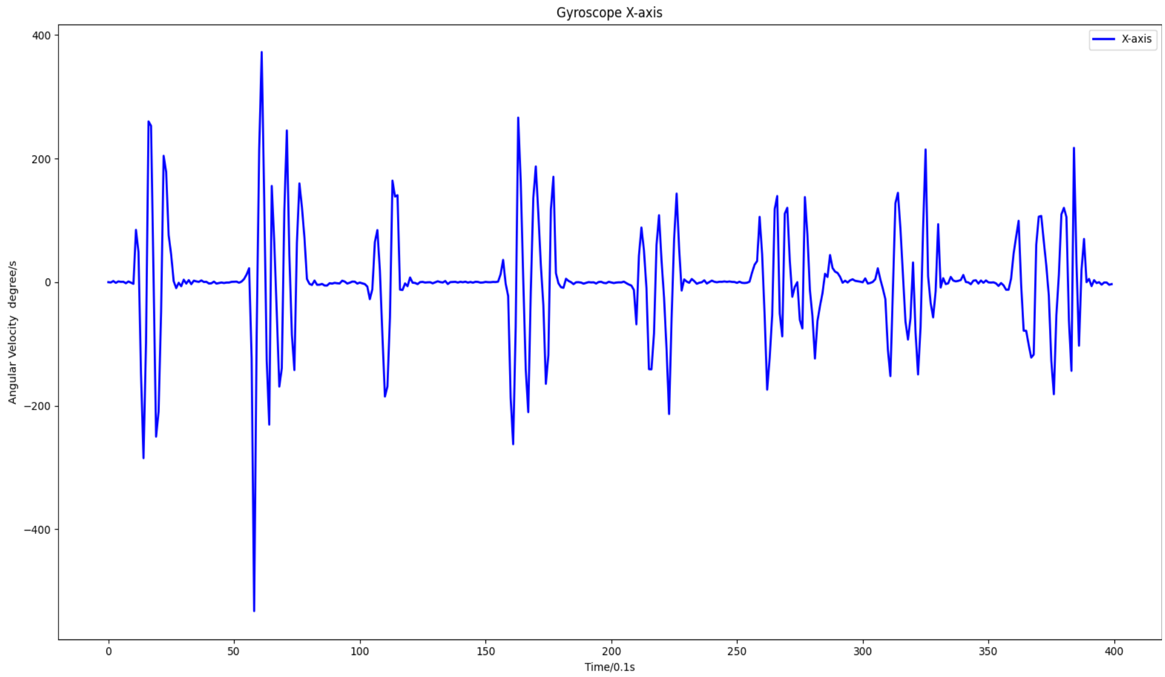


Fig. 3.16 Gyroscope X-axis (good performance axis)

3.4 System Optimization

For the DTW method, the most critical part for recognition accuracy is the quality of templates and samples, because it directly determines the matching degree between template sequences and sample sequences. After considering this character and the data characteristics of IMU, we design two schemes to improve the recognition accuracy, the idle-cutter and the multi-template system.

In practical application and experiment scenarios, there are idle parts before the start point of an air-writing movement and after the end of the movement. If we slice the air-writing samples sequences by a fixed time, every sample will include the idle time, which will reduce the recognition accuracy and generate more time consumption by recognition algorithms. After considering IMU will output accelerometer values directly, we design the idle-cutter that will find the start point and endpoint automatically, then cut the idle time and the remaining sequences only include the air-writing information without idle information. Fig. 3.17 shows waveforms of the X-axis in the accelerometer when air-writing three "A", and after processed by idle-cutter, only these waveforms which include air-writing information in black frames will remain. The operating principle of the idle-cutter as follows. The algorithm calculates the slope range (fluctuation range) of slight involuntary shakes of the hand by accelerometer values from IMU, then calculates the sample slope by X-axis accelerometer values that include three time points (sampling rate is 10 times points per second). If the sample slope is included in the range of hand slight shakes, the algorithm will decide this data is still in idle time. Otherwise, the algorithm will decide the air-writing gesture has already begun and found the start point of an air-writing movement. The endpoint will be determined by the same principle.

In actual experiments, there are considerable variations of results between air-writing samples of the same letters by the same user collected at different times. Eventually, we determine that this issue was caused by different angles (shown in Fig. 3.18) of the user's hand when doing air-writing. This issue reduces the recognition accuracy because the mismatching between template and sample will affect the DTW algorithm. To solve this problem, a multi-template system was proposed which will match the air-writing hand angle of template sequences and sample sequences and then choosing the best group of templates. The matching principle as follows. Firstly, the 3-axis gyroscope will output the hand angles that relative to the horizontal plane. As shown in Fig. 3.19, for 15° to 35° of the air-writing hand angles, samples will match with the template group that was recorded at 25° and then starting recognition algorithms. Samples from 35° to 55° will match with the template group at 45°, and those from 55° to 75° will match with 65° templates. As for the data that hand angle is below 15° or above 75°, people hardly do air-writing in these extreme ranges, so these ranges will not be taken into consideration. The multi-template system provides a better solution for users' different air-writing postures, and it also improves the recognition accuracy because of the better match quality.

The proposed system uses a 9-axis IMU, which includes a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. All 9 axes provide movement data for the DTW algorithm, and some axes also provide data for special functions. The X-axis of the

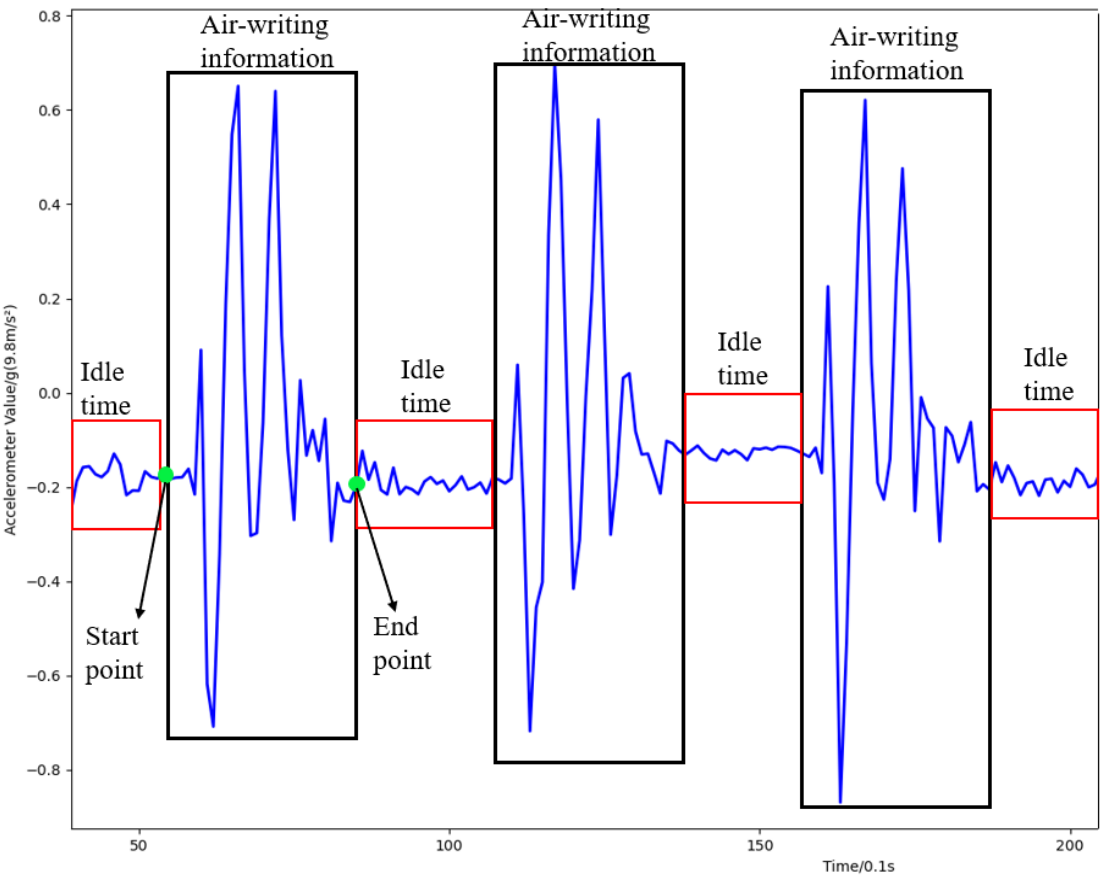


Fig. 3.17 Three waveforms of letter 'A'. Idle-cutter will cut idle time in red frames and only remian useful information in black frames.

Fig. 3.18 Different air-writing hand angles respectively are about 90°, 60°, 30°, and 0° relative to the horizontal plane.

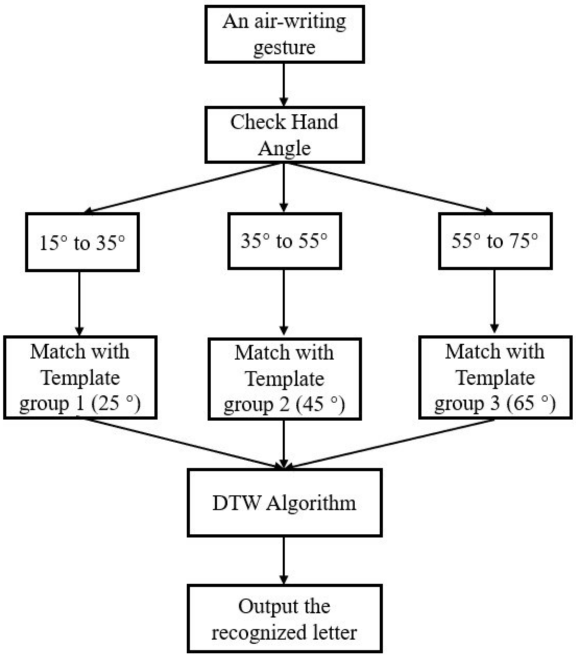


Fig. 3.19 The process of multi-template system.

accelerometer provides data for the idle-cutter, which will use movement data to calculate the sample slope range, then determine the start point and endpoint. The X-axis of the gyroscope directly provides the hand angle values relative to the horizontal plane when doing air-writing, and hand angle values will be used to match the best template group for the DTW algorithm in the multi-template system. Fig. 3.20 shows the collaboration of these 9 axes in a single recognition process.

In the proposed air-writing system, a template includes 9-axis movement data without idle information and a hand angle; and a sample usually includes 9-axis whole movement data. After processing by the idle-cutter and matching with the best template group, every sample axis of IMU will calculate by the DTW algorithm, with the matched axis in the template. In our experiment, a template group usually includes 36 letters and numbers (uppercase alphabet and 10-digits), so every axis of the sample sequence will process by DTW 36 times, then output the most similar letter or number. At last the recognition algorithm will integrate the results of

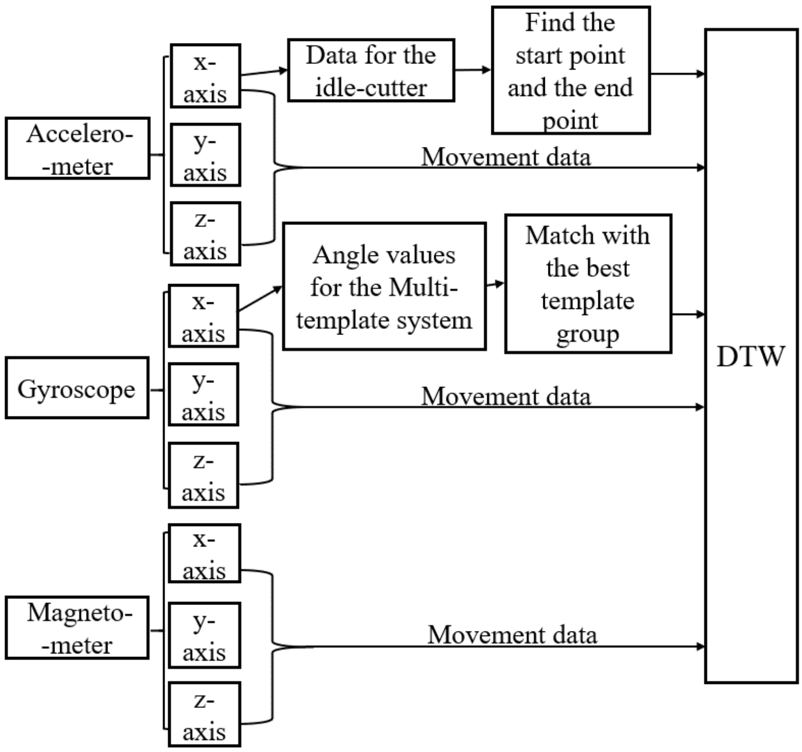


Fig. 3.20 Collaboration of IMU 9 axes.

nine axes together, then output a particular letter or number as the final recognition result which has 5 or more axes successfully match with template sequences. Cooperative work of 9-axis can guarantee high recognition accuracy and improve the system robustness greatly.

In DTW-KNN algorithm, the idle-cutter will be used but the multi-template system will not be used, because we have a much more efficient and rigorous method, KNN algorithm. In a sense, DTW-KNN algorithm can be regarded as a more scientific and effective extension of a multi-template system. In Peak-number algorithm, the idle-cutter is unnecessary, because idle time does not affect our peak selection, and it takes almost no computation.

3.5 Time Consumption

The purposed air-writing system is to be applied to real application scenarios, therefore, in addition to the accuracy rate, the time consumption required by the system from input to output is also an important indicator to measure whether the proposed system conforms to real-time recognition. In many similar gestures recognition research, researchers are committed to real-time recognition. For example, in [20], the author proposed an electromyography (EMG)-based method for gesture recognition, and their real-time standard is "A gesture must be recognized in less than 300 ms". In vision-based methods, researchers also aim to real-time recognition. In [21], the researchers realized a constant processing speed of 30 (fps) for real-time performance. As we have mentioned before, the DTW algorithm and DTW-KNN algorithm mainly use all the information of the IMU data and calculate the distance. Not surprisingly, this kind of thinking will take a lot of time in the calculation. In particular, the optimization scheme we proposed requires more computation time. The Peak-number algorithm only cares about the peak point of time series and does not involve very complex calculation, because this algorithm does not have the problem of cross calculation between different time series, but only focuses on individual time series. According to experience, the peak searching algorithm is relatively mature and fast. The specific time consumption of each algorithm will be described in detail in Chapter 4.

Chapter 4
Experiment and Result
4.1 DTW based Experiment

To evaluate the proposed air-writing system, we wear the IMU to write uppercase letters from “A” to “Z” in the air. To test the robustness of the air-writing system, users are required to deliberately write letters at different speeds. Every air-writing sample will last for 5 seconds, and generally, users will finish a letter between 1.5 and 3.5 seconds. The idle-cutter will slice the 5 seconds sample into a final sample which only includes air-writing information. Besides, participants can use their own writing ways and postures. We conducted three experiments involving a single user’s recognition, different users’ across recognition, and a beginning user’s recognition. All experiment participants are in the 20s, right-handed.

A. Experiment I: Single user

This experiment uses the templates and samples from the same participant. The participant is required to record 3 template groups for the multi-template system and DTW algorithm, and then the participant will start writing letters in the air. From "A" to "Z", "0" to "9", the English alphabet and ten-digits will be written repeatedly 12 times.

The accuracy of letters is 84.6%, and the accuracy of numbers is 98.2%. So, the total air-writing recognition accuracy is 88.4%. The recognition result of every letter as Table. 2 shows. Half of letters have a 100% successful recognition rate and all numbers have 100% accuracy except two “2” were recognized as “7”. The experiment result shows that the proposed air-writing system has high total accuracy in personalized and user-dependent recognition scenarios.

We also test the recognition accuracy without the proposed adjustment system (include the idle-cutter and the multi-template system, stated in Chapter 3). The recognition accuracy of the

Table. 2 The result of DTW-based method in single user case, Experiment I

Letter	**Accuracy (%)**	**Wrongly Recognized as**	**Letter**	**Accuracy (%)**	**Wrongly Recognized as**
A	100		S	100	
B	100		T	33.3	F,O,V, X
C	100		U	50	A, V
D	100		V	83.3	W
E	83.3	I	W	100	
F	100		X	100	
G	100		Y	83.3	G
H	66.7	D, K	Z	100	
I	83.3	A	0	100	
J	100		1	100	
K	75	N, Q	2	83.3	7
L	50	S	3	100	
M	100		4	100	
N	100		5	100	
O	66.7	I, Q	6	100	
P	83.3	I	7	100	
Q	66.7	U	8	100	
R	75	P, Q	9	100	
Accuracy of letters		84.6%	Accuracy of numbers		98.3%
Total accuracy			88.4%		

alphabet letters (“A” to “Z”) is about 64%. It confirmed that the proposed adjustment system plays a significant role in the air-writing system.

B. Experiment II: Fixed template for different users

This experiment uses fixed template groups which have used in Experiment I, and users just need to provide samples. Because fixed template groups are recorded by cursive, so participants in Experiment II will also do air-writing by cursive. Then we use the fixed template with samples written by 3 different participants. This experiment is to verify the performance of the system in user-independent scenarios. Fig. 4.1 shows this process of Experiment.

The letters recognition accurate rates from 3 different users respectively are 4%, 4.8%, 7%. It means the proposed air-writing system could hardly recognize letters if templates and samples are from different users. Besides, if we use their own data as templates for these 3 users, the air-writing system will have good accuracy as Experiment I. These results show that the proposed air-writing system is a high user-dependent system.

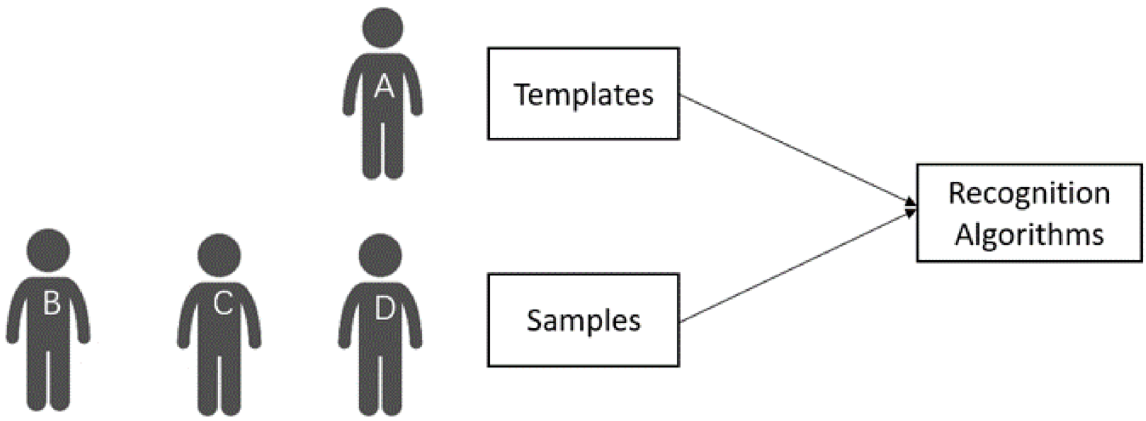


Fig. 4.1 Process of Experiment 2.

C. Experiment III: For a beginning user

In Experiment I and Experiment II, users are familiar with the IMU device and air-writing, because before record data, they have enough time to practice air-writing and cursive. To verify the extreme case, the user of Experiment III is not an English native speaker and firstly contact with cursive write way and IMU device. This user is only required to record a group template, which means the multi-template system will not work, and repeat the air-writing of the alphabet 3 times without any cursive practice. In summary, we test the air-writing system by a very beginning user with a small number of samples.

The total letters' accurate rate of the beginning user is 73%. This result shows that the proposed air-writing system has good performance in extreme cases. This experiment also shows the proposed system will be a competitive scheme when the sample size is limited or there is no enough training time.

4.2 DTW-KNN based Experiment

In the Chapter 3, we have elaborated the basic principle of DTW-KNN algorithm. And there are two key parameters for this algorithm: K value and Axes numbers. Undoubtedly, the K value will affect the recognition accuracy of our proposed air-writing system because of KNN algorithm characteristics, and theoretical analysis is also elaborated in the Chapter 3. The wearable IMU can output 9 axes data, but do we need to use all the axes to get good enough results, just like in the classical DTW algorithm.

Aiming at these two key parameters and following the scientific experimental concept of control variables, we designed two experiments. These data sets of the two experiments were from the same user, because our goal is to compare results with Experiment I: Single user, which has the highest accuracy in DTW based method.

A. Experiment IV: Fix the K value, change axes number

We will change the number of axes in the input DTW-KNN algorithm, at the same time, the K value will be fixed at 5. This experiment uses the same test dataset as shown in Chapters 4.1, 12 groups air-writing alphabets. However, this data set does not include numbers, because in the DTW based method, we found that the recognition of numbers is very simple, with almost a 100% success recognition rate. Therefore, the recognition of numbers will not be carried out in Experiments for Chapter 4.2 and 4.3. This means that after that, what we use the word "accuracy" only for 26 uppercase English letters. In this experiment, we will change the number of input axes, specifically, axis 9 means using all IMU data, axis 6 does not include the magnetometer, axis 3 does not include the magnetometer and gyroscope, and axis 2 and 1, all

Table. 3 The result of DTW-KNN based method when fixed K =5.

K Value	Number of Axes	Accuracy (%)	Time (s)
5	9	100	20
5	6	100	13
5	3	92.3	7.0
5	2	84.6	4.5
5	1 (X-axis)	84.6	2.2
5	1 (Y-axis)	69.2	2.1
5	1 (Z-axis)	88.5	2.2

use the data only from the acceleration. In addition, according to the principle described in the third chapter, the core of DTW-KNN algorithm is to use the similarity of DTW algorithm to replace Euclidean distance. DTW algorithm involves two time series, one is a template, the other is a sample, so in DTW-KNN algorithm, we will record six additional groups of templates (each group includes 26 English letters) for DTW calculation. It is worth noting that the number of template groups should not be less than k, otherwise it may lead to algorithm overlap or inconsistent. Table. 3 shows the result of Experiment IV. From the results, we can find that when k value is fixed, the recognition accuracy is directly proportional to the number of input axes. Especially, when the number of axes is 6 or more, the recognition rate of 26 English letters is 100%. Compared with the classical DTW algorithm, the recognition rate of letters is 84.6%, which is a great progress that means that the recognition is almost error-free. We also notice that when the number of input axes is very small (including 2 axes and 1 axis), the recognition rate is still good, but the system is likely to be unstable, because the amount of data is too small, which leads to dramatic fluctuations in the recognition accuracy. We also found that DTW-KNN is a brute force algorithm that uses a lot of recognition time. The time consumption is directly proportional to the amount of data input. When the maximum amount of data input is 9 axes, it takes nearly 20 seconds to recognize a letter.

B. Experiment V: Fix the axes number, change K value

We will change the K value that is the most crucial parameter in DTW-KNN algorithm, at the same time, the number of axes as input will be fixed at 3 (only use data from accelerometer). The test dataset used is the same as the previous experiments. Table. 4 shows the result of Experiment V.

From the results, we can see that K value has a great influence on the recognition accuracy. High K value brings more computation and fault tolerance, thus ensuring a higher accuracy. And when the K value is too low (i.e. K = 2), it is likely to lead to the collapse of the model and the precipitous decline of the recognition accuracy, resulting in our system cannot complete the recognition task. But from the point of view of time consumption, the influence of K value on recognition time is very small, only less than 10%. Specific analysis of time consumption will be described in Chapter 4.4.

Table. 4 The result of DTW-KNN based method when fixed number of axes is 3.

K Value	**Number of Axes**	**Accuracy (%)**	**Time (s)**
5	3	92.3%	7.02
4	3	84.6%	7.28
3	3	80.7%	7.19
2	3	46.1%	6.91

4.3 Peak-number based Experiment

The Peak-number algorithm has been clearly claimed in Chapter 3, but for the actual experiment, we have some specific rules:

1. Every letter has 3 peak number information because 3-axis gyroscope, but usually we only select 2 axes as the key axis that means the information is necessary. Only when the key axis information of two letters is the same, the algorithm will recognize letters according to the remaining axis.
2. The peak number information will consist of three two-digit numbers, they represent the X, Y, Z-axes of the gyroscope in turn. The first digit of the two-digit number represents the upward peak and the second digit represents the downward peak. The specific rules are shown in Table. 5. Red means key axes. Key axes are selected by many repeated experiments, it can give the system fault tolerance and improve the robustness of the proposed air-writing system.

Table. 5 The representation method of the result.

Example, Letter 'A'	Number of upward peaks	Number of downward peaks
Gyroscope X-axis	2	2
Gyroscope Y-axis	0	1
Gyroscope Z-axis	2	2
Representation method	22 01 22	

The result of Peak-number based experiment is shown in Table 6. We use specific representation method in rule 2. Compared with the result of classical DTW algorithm, the recognition accuracy of this Peak-number method is reduced by 11%. Compared with the DTW-KNN algorithm, the recognition accuracy of this Peak-number method is reduced by 18.4%. However, the Peak-number based method has a crucial advantage: Very short time consumption.

Table. 6 The result of Peak-number algorithm.

Letters	Peak number	Accuracy (%)	Letters	Peak number	Accuracy (%)
A	22 01 22	100%	N	22 11 12	75%
B	12 21 11	66.7%	O	11 10 11	83.3%
C	01 10 11	91.7%	P	11 11 11	75%
D	12 11 11	100%	Q	12 20 22	66.7%
E	12 22 23	50%	R	12 32 12	91.7%
F	11 11 12	83.3%	S	11 11 21	66.7%
G	02 11 22	66.7%	T	01 10 11	58.3%
H	12 22 01	58.3%	U	12 10 11	75%
I	01 10 02	83.3%	V	11 10 01	75%
J	11 11 21	58.3%	W	22 10 00	66.7%
K	12 21 12	66.7%	X	12 11 11	92%
L	01 10 01	100%	Y	01 10 01	41.7%
M	23 32 12	66.7%	Z	11 21 22	100%
Total		**75.3%**			

4.4 Result Summary

Table. 7 shows the results of different experiments and their recognition time.

The conclusion in Table 5 is obvious, each algorithm has its advantages and disadvantages. The classical DTW algorithm has a good performance in the case of user-dependent, and a balance between accuracy and time consumption is achieved. DTW-KNN algorithm sacrifices the recognition time to obtain higher accuracy. It can be regarded as an upgraded algorithm of DTW algorithm in terms of computation and complexity. The advantage is a very high accuracy, Table. 7 Summary of different experiments.

Methods	Accuracy	Time Consumption
DTW	84.6%	8.1 s
DTW-KNN (K=5, 9 axes)	100%	20 s
DTW-KNN (K=5, 6 axes)	100%	13 s
DTW-KNN (K=5, 3 axes)	92.3%	7.0 s
Peak-number algorithm	75.3%	0.0003 s

and can even reach 100% when enough data is used, but the time consumption is very long, and if too much input data is reduced, the recognition system will be lack of robustness and easily collapse. Finally, the accuracy of this algorithm is relatively low compared with the first two, but it is only about 10% lower, but it hardly needs recognition time (0.0003s), because the peak searching algorithm is very fast and mature.

The ultimate goal of the proposed air-writing system is to use in the actual use of the scene, to help users or disabled people quickly and efficiently non-contact text output. So in addition to accuracy, which is very important for all pattern recognition tasks, real-time recognition is another equally important indicator for our system. Obviously, the two methods based on DTW and DTW-KNN algorithms are difficult to achieve real-time recognition, because they need a lot of data calculation to ensure the accuracy. And it is very difficult to optimize the time consumption. In fact, we have used the optimized FastDTW. Our proposed Peak-number algorithm is very suitable for real-time recognition tasks. Although its accuracy may not be enough for industrial products, there is still a great potential for optimization. The optimization of accuracy is much simpler than the optimization of time consumption. To sum up, we think that the proposed Peak-number algorithm is very effective and has potential for the real-time recognition task of air-writing.

Chapter 5
Conclusion and Future Plan
5.1 Conclusion

This paper presents an air-writing system based on an inertial measurement unit, And the wearable ring is designed based on IMU. The air-writing system uses DTW, DTW-KNN and Peak-number algorithms as the main recognition algorithms to do experiments. After consideration of DTW characters, an adjustment system has been proposed to improve the recognition accuracy and to minimize the recognition time consumption. The idle-cutter in adjustment system also gives a solution for the problem that the acceleration caused by the user's unconscious hand tremble might significantly affect gesture recognition results, which has influenced some researchers in gesture recognition works. Besides, the proposed multi-template system also gives a new idea to optimize the DTW application, and this idea is not limited to the hand angle parameter and gesture recognition field.

We creatively put forward the peak number algorithm, a new way to solve the problem of time series analysis. In the experiment, we have done experiments in three directions: DTW, DTW-KNN, and Peak-number based method. Experiment results show that in the user-dependent and personalized case, the proposed air-writing system has good performance and high accuracy (84.6%). For extreme conditions, especially under the limitation of time and sample size, the DTW based system will be a competitive choice. Besides, the experiment results also show that user-independent across movement gesture recognition only based on the DTW algorithm is hard to achieve. We creatively combine DTW and KNN algorithms, the core idea is to use DTW distance to replace Euclidean distance in KNN. As an upgrade of DTW algorithm in computation, DTW-KNN algorithm has made great progress in accuracy. It can even achieve 100% accuracy, which means that the recognition is almost error-free. We also discuss the K value and the number of axes of the algorithm under the control variable idea, and draw a series of meaningful conclusions. However, the cost of increasing the amount of computation is that the time consumption becomes very long, as a result, this method is difficult to be used in real-time recognition. The proposed Peak-number algorithm performs very well in time consumption, but the accuracy (75.3%) is about 10% - 18% lower than DTW and DTW-KNN, but it also has great potential to improve. And the algorithm only needs a very short recognition time (0.0003 seconds), so we believe that this method will be brilliant in real-time recognition in the future.

5.2 Challenges and Future Plan

From the conclusion of the experiment, no matter which algorithm is used, the accuracy of the system still has a large optimization space. And in the actual application scene, there are many very complex details to deal with, such as the meaningless movement of the user's hand, or the motion connection between letters and so on. Our goal is that users can recognize real-time and efficient air writing, and aim at complete sentences and words, not just letters. This is a challenging but meaningful work.

For the classical algorithm DTW, it is very difficult to make considerable optimization in terms of accuracy or time consumption. We mainly consider the optimization of the proposed Peak-number algorithm in the future. Specifically, we will optimize it from three aspects:

1. Key axes. We propose key axes to do some non-strict one-to-one fuzzy recognition, so just like the Multi-template system we proposed, we will design multiple correct peak number information groups to correspond with the input samples. This will improve the overall recognition accuracy in non-laboratory application scenarios.

2. Adopt some machine learning ideas. We mainly want to design a network based on the idea of Generative Adversarial Networks (GANs) [19], including generator and discriminator, so that the network can train itself and adjust the parameters of the peak number algorithm, to obtain better accuracy.
3. Expand the system and build more powerful algorithms to recognize complete words and sentences, not limited to letters.

For wearable hardware design, wireless is very important. We almost use Bluetooth or other wireless communication methods to connect wearable devices and terminals (mobile phones or computers). In addition, improving the robustness of the system is also an important direction, because there may be some angle or position changes in the actual use.

Research Achievements

Y. Luo, J. Liu and S. Shimamoto, "Wearable Air-Writing Recognition System employing Dynamic Time Warping," *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021, pp. 1-6, doi: 10.1109/CCNC49032.2021.9369458.

Reference

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

[2] T. Lu, "A motion control method of intelligent wheelchair based on hand gesture recognition", Proc. 8th IEEE Conf. Ind. Electron. Appl. (ICIEA), pp. 957-962, Jun. 2013.

[3] 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.

[4] S. Lian, W. Hu and K. Wang, "Automatic user state recognition for hand gesture based low-cost television control system", IEEE Trans. Consum. Electron., vol. 60, no. 1, pp. 107-115, Feb. 2014.

[5] C. Z. Qu, D. Y. Zhang and J. Tian, "Online Kinect Handwritten Digit Recognition Based on Dynamic Time Warping and Support Vector Machine", ournal of Information & Computational Science, vol. 12, no. 1, pp. 413-422, Jan. 2015.

[6] M. Chen, G. AlRegib and B. Juang, "Air-Writing Recognition—Part I: Modeling and Recognition of Characters, Words, and Connecting Motions," in IEEE Transactions on Human-Machine Systems, vol. 46, no. 3, pp. 403-413, June 2016.

[7] 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.

[8] 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.

[9] Y. Zhou, Z. Dai and L. Jing, "A Controlled Experiment Between Two Methods on Ten-Digits Air Writing," 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, 2016, pp. 299-302.

[10] T. Pan, C. Kuo, H. Liu and M. Hu, "Handwriting Trajectory Reconstruction Using Low-Cost IMU," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 3, no. 3, pp. 261-270, June 2019.

[11] T. Tsai and J. Hsieh, "Air-writing recognition using reverse time ordered stroke context," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, 2017, pp. 4137-4141.

[12] P. Wang et al., "A Gesture Air-Writing Tracking Method that Uses 24 GHz SIMO Radar SoC," in IEEE Access, vol. 8, pp. 152728-152741, 2020.

[13] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", Ieee Transactions on Acous-Tics Speech and Signal Processing, vol. 16, pp. 43-49, 1978.

[14] Salvador, Stan, and Philip Chan. "Toward accurate dynamic time warping in linear time and space", Intelligent Data Analysis, vol. 11, no. 5, pp. 561-578, Oct. 2007.

[15] M. A. Imtiaz and G. Raja, "Isolated word Automatic Speech Recognition (ASR) System using MFCC, DTW & KNN," *2016 Asia Pacific Conference on Multimedia and Broadcasting (APMediaCast)*, 2016, pp. 106-110, doi: 10.1109/APMediaCast.2016.7878163.

[16] FIX, Evelyn; HODGES JR, Joseph L. Discriminatory analysis-nonparametric discrimination: Small sample performance. CALIFORNIA UNIV BERKELEY, 1952.

[17] COVER, Thomas; HART, Peter. Nearest neighbor pattern classification. IEEE transactions on information theory, 1967, 13.1: 21-27.

[18] Ventzas, Dimitrios & Petrellis, Nikos. Peak Searching Algorithms and Applications. Proceedings of the IASTED International Conference on Signal and Image Processing and Applications, SIPA 2011. 10.2316/P.2011.738-049.

[19] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In NIPS'2014.

[20] A. G. Jaramillo and M. E. Benalcázar, "Real-time hand gesture recognition with EMG using machine learning," *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, 2017, pp. 1-5, doi: 10.1109/ETCM.2017.8247487.

[21] T. Kirishima, K. Sato and K. Chihara, "Real-time gesture recognition by learning and selective control of visual interest points," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 351-364, March 2005, doi: 10.1109/TPAMI.2005.61.

Appendix I
DTW, Fast DTW and *c*DTW*w*

The Dynamic Time Warping is used as the best method in time series analysis in many domains. Original DTW algorithm was proposed very early and it is generally believed by thousands of researchers that this algorithm is exact but very slow and needs optimization. Against this background, FastDTW has been put forward as a kind of "fast" algorithm, and many researchers have accepted it and used it instead of the original DTW, especially in the fields that are not familiar with computer algorithms, such as engineering, medical, communication and so on.

But a recent study has come to some surprising conclusions, in [A.1], the authors Renjie Wu and Eamonn J. Keogh said that FastDTW is approximate and generally slower. It means FastDTW may decrease the accuracy of our experiments and use more recognition time. They also introduce *c*DTW*w* (DTW with the constraint). In short, *w* means the maximum percentage of algorithm warping allowed, as Fig. A.1[A.2] shows, *w* is r/n (From 0 to 100).

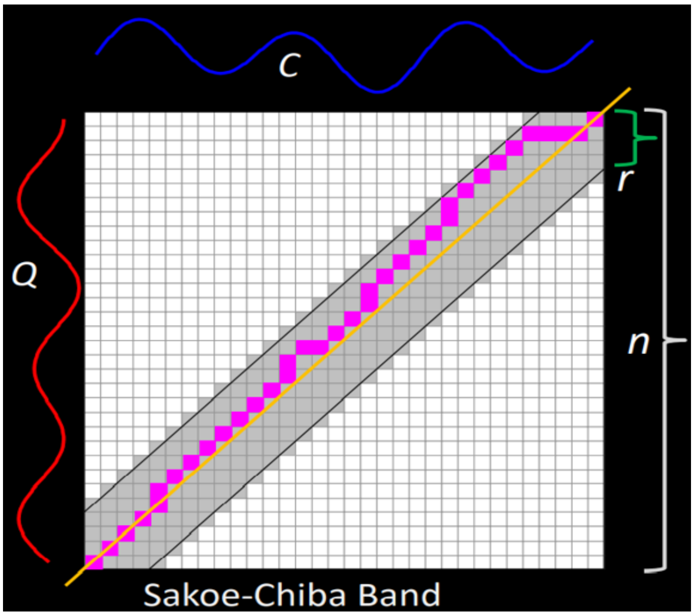


Fig. A.1 Value of *w*

I have tested the time consumption for two long time series, the two sequences are data for my IMU devices, and both have 7800 data (means 780s). Table. A.1 shows the result. We can find

Algorithms	Time Consumption(s)	DTW Distance
cDTW$_{100}$	2.3	11768656
FastDTW	4.6	170730
DTW	320.9	169633

Table. A.1 Results of two time series by DTW, FastDTW and *c*DTW

it clearly that FastDTW is faster than DTW, that's why thousands of researchers believe it. For mature algorithms such as DTW, many researchers tend to call their functions in package directly instead of making their own wheels, it's a kind of trust in the community. So, I speculate that there may be some problems in the source code of the DTW Python package, which makes it slow and gives many researchers wrong conclusions. Besides, we can find that the DTW distance of cDTW$_{100}$ is different from FastDTW and DTW, the reason is in *c*DTW, every time we calculate the distance between two points, we do less square root operation. It's a good operation trick that because in pattern recognition, as long as we use the same set of rules, the result will not be wrong. And obviously, the lack of square root operation will lead to less computation time.

For w value, it is clear that w value is positively correlated with calculation time. However, a larger w value does not mean higher accuracy. Some results of real experiments in my air-writing system are shown in the Table. A.2. The dataset includes a template and 6 samples,

Algorithms	Time Consumption (s)	Accuracy (%)
FastDTW	8.3	80.1
cDTW$_{30}$	1.9	76.9
cDTW$_{10}$	1.55	81.4
cDTW$_{5}$	1.45	82.7

Table. A.2 Results of real experiments.

both include 26 alphabet letters. It is clear that *c*DTW greatly reduces the recognition time, and the appropriate *w* value will result in a higher recognition rate than FastDTW.

To a simple conclusion: We believe that *c*DTW*w* is a better algorithm compares with FastDTW, it can greatly reduce the recognition time and also has better performance in accuracy. The value of *w* is a very crucial part for accuracy. However, for our proposed air-writing system, because of the huge amount of computation, the recognition time still over 1 second that make it can not achieve real-time recognition.

I would also like to express my sincere thanks to Prof. Keogh. He read my paper which published in CCNC2021 and send an email to me to show their new work in DTW and suggest me to use *c*DTW*w*. I also suggest that researchers who have read this thesis consider using *c*DTW, which is really a good method in time series analysis.

Reference (Appendix)

[A.1] R. Wu and E. J. Keogh, "FastDTW is approximate and Generally Slower than the Algorithm it Approximates," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2020.3033752.

[A.2] Abdullah Mueen, Eamonn J. Keogh: Extracting Optimal Performance from Dynamic Time Warping. KDD 2016: 2129-213

Appendix II
Source Code

I'm confident of the repeatability of the proposed air-writing system. Here is the source code of the system, please mark the source if you want to use or reprint it. There are mainly 3 parts, if you want to automatically loop through all the results, you must write an all function that synthesizes the 3 parts and more details in the code is needs.

1. DTW:

Package:

```
from scipy.spatial.distance import euclidean
from fastdtw import fastdtw
import timeit
import heapq
import math
import numpy as np
```

IMU data reading:

```
    def txtinput1(n, X=[]):
        with open(r'C:\Users\xxxxx\xxxxx\samplexxx.txt', 'r') as f:
            lines = f.readlines()
            for line in lines:
                value = [float(s) for s in line.split()]
                X.append(value[n])
        return X
```

Using DTW:

```
def recognition(n, T):
```

```
    SAM = txtinput1(n)
        sample = SAM[z:x]
        TEM = txtinput2(n)
        times = 0
        Y = []
        while times < T:
            template = TEM[0 + (50 * times):49 + (50 * times)]
            dis, path = fastdtw(sample, template, dist=euclidean)
            Y.append(dis)
            times = times + 1
        '''print(Y)'''

        return Y
        recogresult = dtwresult.index(min(dtwresult))
        return recogresult
```

Synthesize the nine axis results:

```
# Re 9-axis
    axises = (0, 1, 2, 3, 4, 5, 9, 10, 11)
    axisresult = []
    for axis in axises:
        axisresult.append(recognition(axis))
    result = max(axisresult, key=axisresult.count)
    allresult = []
    allresult.append(result)
    return allresult
```

2. KNN

```
def KNN_DTW(n,z,x):
    def all(n, T):
        def txtinput1(n, X=[]):
            with open(r'C:\Users\Lenovo\Desktop\26letterX1.txt', 'r') as f:
                lines = f.readlines()
                for line in lines:
                    value = [float(s) for s in line.split()]
                    X.append(value[n])
            return X

        def txtinput2(n, X=[]):
            with open(r'C:\Users\Lenovo\Desktop\LUO26X6.txt', 'r') as f:
                lines = f.readlines()
                for line in lines:
                    value = [float(s) for s in line.split()]
                    X.append(value[n])
            return X

        # DTW
        SAM = txtinput1(n)
        sample = SAM[z:x]
        TEM = txtinput2(n)
        times = 0
```

```
    Y = []
    while times < T:
        template = TEM[0 + (50 * times):49 + (50 * times)]
        dis, path = fastdtw(sample, template, dist=euclidean)
        Y.append(dis)
        times = times + 1
    '''print(Y)'''

    return Y

# Get k of smallest dtw distance
m = all(n, 156)
min_number = heapq.nsmallest(6, m)
min_index = []
for t in min_number:
    index = m.index(t)
    min_index.append(index)
    m[index] = 0
'''print(min_number)'''
'''print(min_index)'''
result = heapq.nsmallest(2, min_index)
'''print(result)'''
# label, 1 is A, 26 is Z
n = 0
final = []
```

```
    while n < 2:
        mathres = math.ceil((result[n] + 1) / 6)
        n = n + 1
        final.append(mathres)
    maxlabel = max(final, key=final.count)
    return maxlabel
'''print(KNN_DTW(0))'''
```

3. Peak-number method

```
from scipy.signal import find_peaks
#get peak number's information
    peaks1, _ = find_peaks(x[n:n + 50], height=45, distance=8)
    peaks_neg1, _ = find_peaks(-x[n:n + 50], height=45, distance=8)
    peaks2, _ = find_peaks(y[n:n + 50], height=45, distance=8)
    peaks_neg2, _ = find_peaks(-y[n:n + 50], height=45, distance=8)
    peaks3, _ = find_peaks(z[n:n + 50], height=45, distance=8)
    peaks_neg3, _ = find_peaks(-z[n:n + 50], height=45, distance=8)
    print(len(peaks1),len(peaks_neg1))
    print(len(peaks2), len(peaks_neg2))
    print(len(peaks3), len(peaks_neg3))
```