EasyChair Preprint

№ 5994

Continus Finger Tracking System based on Inertial Sensor

Yangyang Fang, Xin He and Qun Fang

July 3, 2021

Continus Finger Tracking System based on Inertial Sensor

Abstract—Finger tracking has become an appropriate approach to interact with smart wearable devices or virtual reality(VR). However, designing circuit system almost required in many system. In this article, we present a continuous finger tracking system, which does not need special equipment or design a special environment. We regard the acceleration sequence of 2.4 seconds before as the features of displacement between current 0.04 seconds. In order to avoiding accumulative errors, caused by double integration,we using long short-term memory(LSTM) models to calculate the displacement directly at the corresponding time. In particular, there is no necessary to know the initial speed in this way. Our system has a resolution of 0.38mm and an accuracy of 2.32mm per frame under 25HZ sampling rate. The system can draw the target track accurately.

Index Terms—inertial sensor, finger tracking, LSTM

I. INTRODUCTION

In recent years, with the rapid development of smart wearable devices such as smartwatches, smart rings, and VR technologies, the interaction methods in a digital scenario have been diversified, not only mouses, keyboards, or a touch screen. Then, many researchers have expanded the interaction ways with smart devices in various methods, especially by tracking fingers. And finger tracking can be divided into two categories: discrete hand posture classification and continuous finger movement trajectory tracking. Early solutions including but are not limited to use sound signals propagate on the skin to track fingers [8], [17]. Nevertheless, it will be interfered with by environmental noise. Magnetic field tracking [11], [27] needs to separate the geomagnetic field. Self-occlusion is a severe interference that must be eliminated for computer vision tracking [29], [30].

In this work, consider the cost and practicability, we attempt to make use of only an inertial sensor for tracking finger. Almost all wearable devices integrate acceleration sensors, so there is no need to design new hardware circuits. Besides, the system can overcome the interference from the environment case the inherent characteristics of the inertial sensor. Different from the traditional finger tracking system based on inertial sensors, to avoid the accumulative errors caused by double integration, we choose the long short-term memory(LSTM) model to calculate displacement. In this case, we train a pipeline model to calculate displacement with acceleration sequences at the corresponding time and then accumulate the displacement at each time to get the motion trajectory. The details will be explained in the Theory of Operation section.

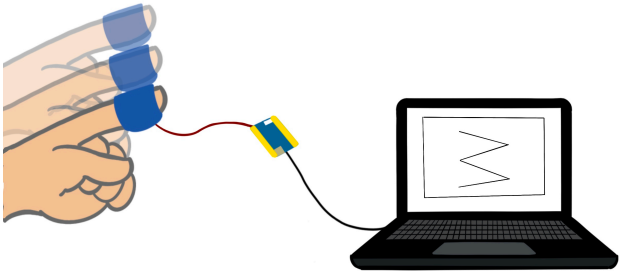


Fig. 1. Finger tracking based on inertial sensor

The main contribution of our work is presenting a new finger tracking system based only on an inertial sensor which avoids accumulative errors caused by double integral. Compared to previous work, our system is more portable and usable. Experimental results show that the average displacement error of each segment calculated by LSTM is less than 2.32mm. By accumulating each segment of displacement, the trajectory of the finger can be completely restored. In the end, we build an application for handwriting letters and numbers.

II. REKATED WORK

In this section, we mainly introduce several important finger tracking methods.

A. Magnetic Tracking

Magnetic field tracking system can be characterized by whether they use alternating current (AC) or direct current (DC) to produce magnetic fields [19]. In DC tracking system, permanent magnets or electromagnets

are matched to generate a constant magnetic field, which can be measured by a magnetic sensor, and continuously interact with smart devices [1], [4], [7], [10], [16]. However, the magnetic field generated by permanent magnets is difficult to be separated from the earth's magnetic field. In this case, many reachers choose AC magnetic fields in recent years [5], [21], [22]. According to Maxwell's equation, an AC magnetic field is generated when an AC electric current passed through a wire coil. Parizi used the fingerprint method to track the magnetic field. The position of the finger is estimated by comparing magnetic filed collected by a magnetic sensor with a database containing magnetic filed data and corresponding position information [19].

B. Computer Vision Tracking

With the growth of computer capability and the extensive application of machine learning, finger tracking based on computer vision has become a viable option. Researchers placed multiple cameras to capture different perspectives of the hand [2], [9], [23], [24]and restored the complete hand model by combining data from different perspectives. The server limitation of this method is self-occlusions that leads to a decrease in the perception of the object [3]. Combined with optical tags, Pavllo used inertial sensors to predict the occluded mark location in 2018 [20].

C. Other Approaches for Finger Tracking

Apart from the tracking methods proposed in the previous section, there are also other methods to track the movement or posture of the fingers. Wang used a radio frequency spectrum for fine-grained hand posture recognization [26]. SkinTrack leverages electrical waveguides to track fingers on the arm [31]. Truong implemented a continuous capacitive sensing system to recognize hand gestures [25]. FingerIO uses cell phone sonar for fine-grained finger tracking [18]. WIFI channel state information(CSI) can also be used to track finger movement [14], [28]. FM-Track uses acoustic signals to track multiple targets [13]. Also, there are still many ways to track hands.

III. THEORY OF OPERATION

In this section, we introduce the theoretical basis of the experiment primarily, explain how to build a pipeline model from acceleration to displacement, and elaborates ultimate goal that we hope to achieve.

If acceleration has been collected, assuming that the acceleration a_t is constant at every time interval. And the initial velocity is v_0. Then we can get this:

$$v_1 = v_0 + a_1 * t \qquad v_{ture1} = v_1 + \delta_1 \tag{1}$$

Here, v_{true} stands for true velocity and δ represents difference value with calculated velocity. The rest can be done in the same manner :

$$\begin{aligned} v_2 &= v_1 + a_2 * t & v_{ture2} &= v_2 + \delta_2 \\ v_3 &= v_2 + a_3 * t & v_{ture3} &= v_3 + \delta_3 \\ &\bullet & & \\ &\bullet & & \\ &\bullet & & \\ v_n &= v_{n-1} + a_n * t & v_{truen} &= v_n + \delta_n \end{aligned} \tag{2}$$

Note that the difference error δ_{n-1} at time n-1 will be accumulated by difference error δ_n of v_n at time n . After multiple accumulations, the difference error between v and the true value v_{true} will be unaccepted, which eventually leads to a complete distortion of the velocity v.

Given that v_t depends on the velocity v_{t-1} at time $t-1$ and the acceleration a_t from time $t-1$ to t. We default to the acceleration from time $t-1$ to time t is constant. Similarly, the velocity v_{t-1} at time $t-1$ depends on v_{t-2} and a_{t-1}. By analogy, v_{t-n+1} relay on v_{t-n} and a_{t+1}. Then, the displacement s at all moments can be obtained through v, which shows in Figure 2(a). In this way, a simplified model diagram can be obtained in Figure 2(b). And the closer acceleration a to time t, the greater the impact on the displacement s_t. Given the time-sequential nature of accelaration, appropriate machine learning models ought to be selected in the experiment or practical application .

Due to their power in modeling the dynamics and dependencies in sequential data [15], LSTM is appropriate to build a pipeline model from acceleration to displacement. Simplify the relationship shows in 2(a), replace the calculation of velocity v_t with the function f, and get a mapping relationship shows in Figure 3. So far, what we hope to achieve is to use the first n acceleration data from the current moment to directly predict the displacement s_t at the current moment, skipping the intermediate calculation of the velocity. Which means that we make use of acceleration $a_{t-n+1} \sim a_t$ from time t-n+1 to time t as the features of displacement s_t at time t, and represent it as input to build the LSTM in Figure4.

The details of the LSTM model parameter setting and data preprocessing will be described in the next section.

IV. LONG SHORT-TERM MEMORY MODEL

In this section, we elaborate on the input and output structure and parameter settings of the LSTM model, as well as data acquisition and preprocessing.

A. Model

As mentioned in the previous section, the system takes the acceleration $a_{t-n+1} \sim a_t$ from time t-n+1 to time t as the features of the displacement s_t at time t. Given the x and y directions , the input end is a dimensional vector as n*2, and the output end is a 1*2 vector. According to the actual finger movement state, the value of n should

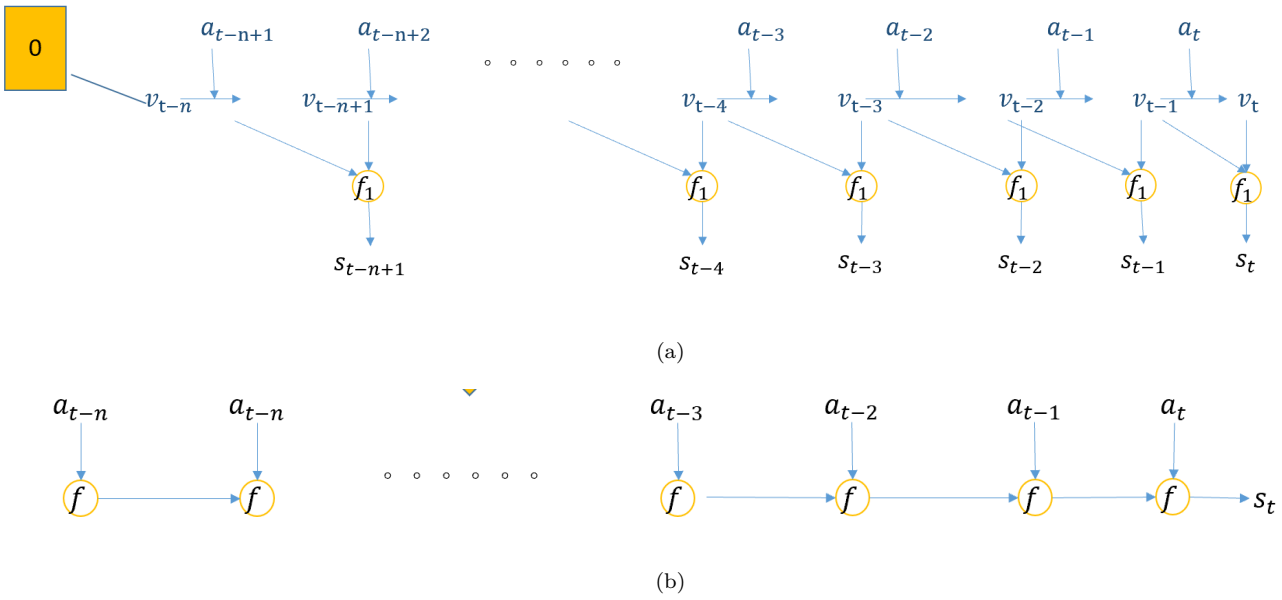


Fig. 2. Model from acceleration to displacement

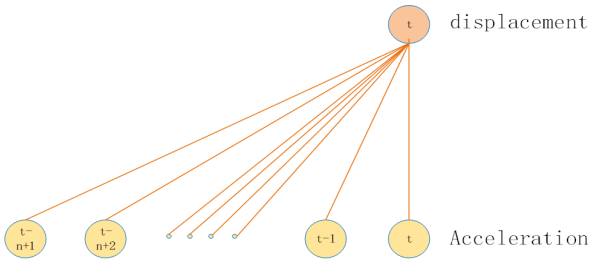


Fig. 3. The mapping from acceleration to displacement: n represents the previous n time nodes, and t represents time. A single node on the upper layer represents the displacement at time t, and the n nodes on the lower layer represent acceleration data from t-(n-1) to time t.

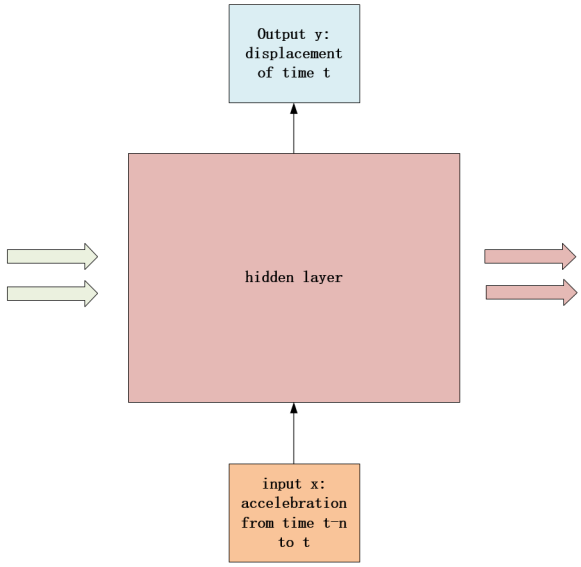


Fig. 4. LSTM schematic diagram: the input data is a two-dimensional vector of n*2, and the output data is the displacement in the x and y directions.

be less than 100, which means that s_t is determined by the acceleration a in the previous 4 seconds. Meanwhile, the sampling rate is 25HZ which means 100 data are sampled in 4 seconds. Aim to choose the value of n, we adjust different values of n into the model training and then calculate the standard deviation. To this end, we collected 15 minutes of motion (writing letters) data, about 22,000 frames of data as the training set, and 45 seconds of motion (writing the same letters) data, about 1,000 frames of data, as the test set. The experimental results are shown in Figure 5. It can be seen from the results that as the value of n increases, the standard deviation shows a downward trend at a sampling rate of 25HZ, the value of n is between 20 and 80. To reduce the time cost of model prediction, 60 is chosen as the value of n. That is, the input of LSTM is a 60*2 vector.

B. Preprocessing

Noise Reduction——It is necessary to eliminate the jitter problem of the data caused by the floating of the LED mark captured by the camera and the inherent characteristic of the inertial sensor. To ensure the portability of the model, we did not preprocess the acceleration, but the displacement data ware denoised

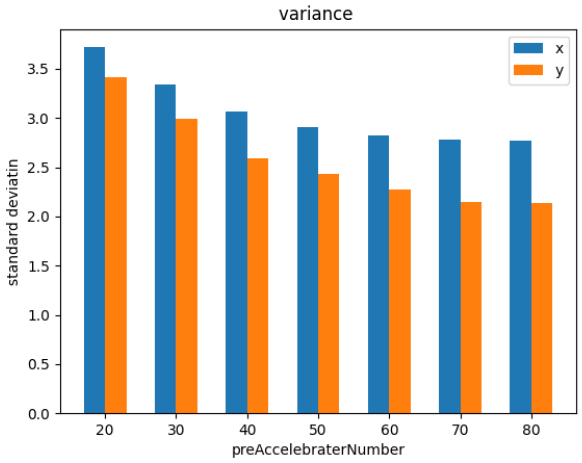


Fig. 5. At the 25HZ sampling rate, the value of n is between 20 and 80. As the value of n increases, the standard deviation shows a downward trend.

to alleviate the problems of jitter and light source drift. Daubechies wavelet is chosen, in which the order of vanishing moment is 4, to complete by decomposing, denoising, and reconstructing.

Synchronization——As there is no synchronization signal to synchronize the acceleration captured by the inertial sensor and the displacement captured by the camera, it is necessary to synchronize the two types of data on PC. The acceleration sequences keep fluctuating within a certain range as the inertial sensor remained stationary in the initial state of the experiment. It is worth noting that the accelerations in the stationary are not zero caused by gravity. And displacement fluctuates around zero within ± 1 pixels because of the error of the position capture algorithm. Once the experimenter wearing an inertial sensor starts to move, the acceleration and displacement will have a sudden change that is defined as the starting time in Figure 6. By setting, the frame rate taken by the camera is 25fps as same as the sampling rate of the inertial sensor. In this way, the obtained displacement sequence and acceleration sequence maintain a one-to-one correspondence. And we get the synchronized acceleration and displacement.

C. Train

We choose average absolute error as the loss function 3 so as to reflect the actual situation of the prediction. Among them, $h(x_i)$ stands for the predicted value of the model, and y is the template value (displacement captured by camera).

$$MAE(X,h) = \frac{1}{m}\sum_{i=1}^{m}|h(x_i) - y_i| \tag{3}$$

The model uses adaptive gradient optimization algorithm 4 to optimize, where g_t represents the gradient of

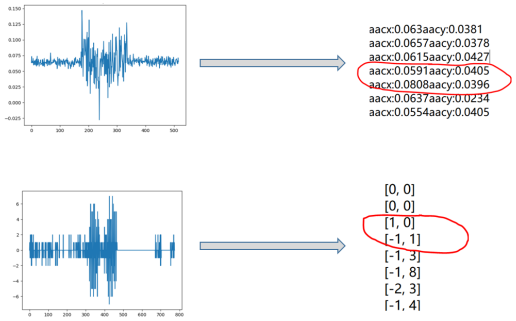


Fig. 6. Find the initial moment through the mutation point

the t-th time step. The specific algorithm description can be found in the article [6], [12].

$$\theta_{t+1} = \theta_t - \eta \bullet g_t / \sqrt{\sum_{i=1}^{t} g_t} \tag{4}$$

V. RESULTS

A. Experimental Settings

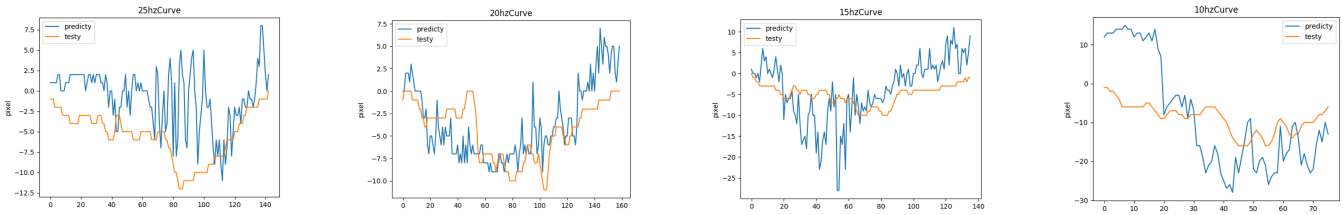
The experiment needs an accelerometer module with an LED mark. Our system uses a commercial module of mpu6050. Acceleration includes the x-axis and y-axis between $\pm 2g$, collected by stm32 development board with an accuracy of four decimal places. And we collect displacement that is an integer that includes the x-axis and y-axis of the led mark through the camera. Input acceleration sequence and displacement sequence into the PC to synchronize the data stream. And train through python. As shown in Figure 8

We recruited 4 participants (3 men and one woman, aged between 20-30). The data collection was carried out in a lab that keeps in the dark to ensure no influence of illuminant. The writing area is about 49.5cm $\times$ 28cm in size. The sampling rate of the inertial sensor is 25HZ, and the frame rate of the camera is 25fps. During operation, the participant wearing the inertial sensor on the index finger of the right-hand remains horizontal and moves slowly on the desktop. Meanwhile, the camera collects displacement on the top.

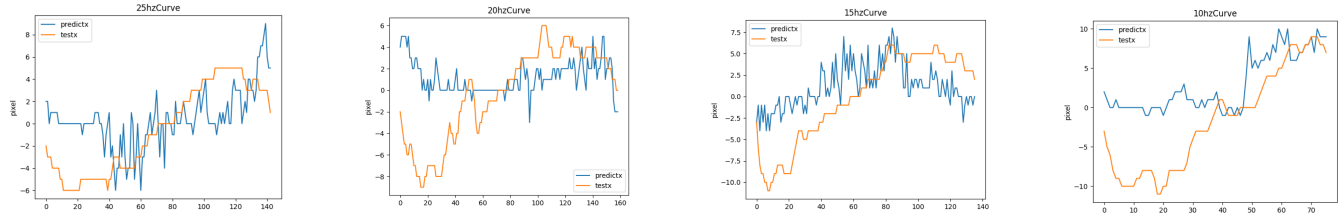
B. Tracking Accuracy

Displacement, which consists of the x-axis and y-axis in each time interval, was used as indicators to evaluate the accuracy. Before the operation, we show the participant drawing target. And then, the participant wears the sensor to draw the corresponding image on the desktop. The drawing speed is roughly kept below 30cm per second if not mentioned.

For the convenience of observation, we expand the displacement between each frame linearly. The ordinate



(a) 25hzY (b) 20hzY (c) 15hzY (d) 10



(e) 25hzX (f) 20hzX (g) 15hzX (h) 10hzX

Fig. 7. The displacement of each time is overlapped according to the starting point. The upper part is the displacement in the y-axis and the lower part is the displacement in the x-axis.

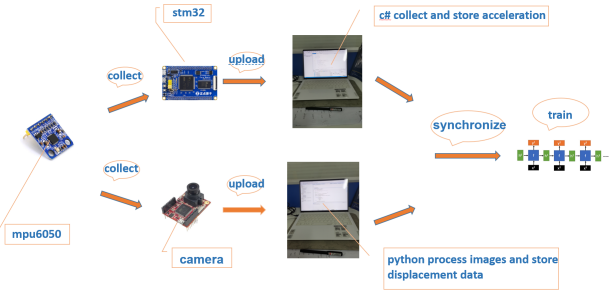


Fig. 8. experiment process

of Fingure 7 represents the displacement, and the unit is pixel. Each pixel equals to 0.3867mm. Then the calculated value (displacement calculated by the pipeline model) and the template value (displacement captured by the camera) are coincident through the synchronized data stream. And then, the standard deviation graph between the template value and the calculated value is calculated. The experimental results show that the standard deviation of displacement in the x-direction and y-direction is less than 6 pixels about 2.32mm at a sampling rate of 25hz.

The influence of sampling frequency——Before the operation start, reachers gave the participant an irregular circle and asked the participants to draw images relatively freely. Participants first draw ten circles as the training set and then draw a single circle as the test set. The same experiment was performed 4 times, sampling at 10HZ, 15HZ, 20HZ, 25HZ shown in Figure9(a). Experimental results show that as the sampling rate increases, the standard deviation of displacement between each period will decrease.

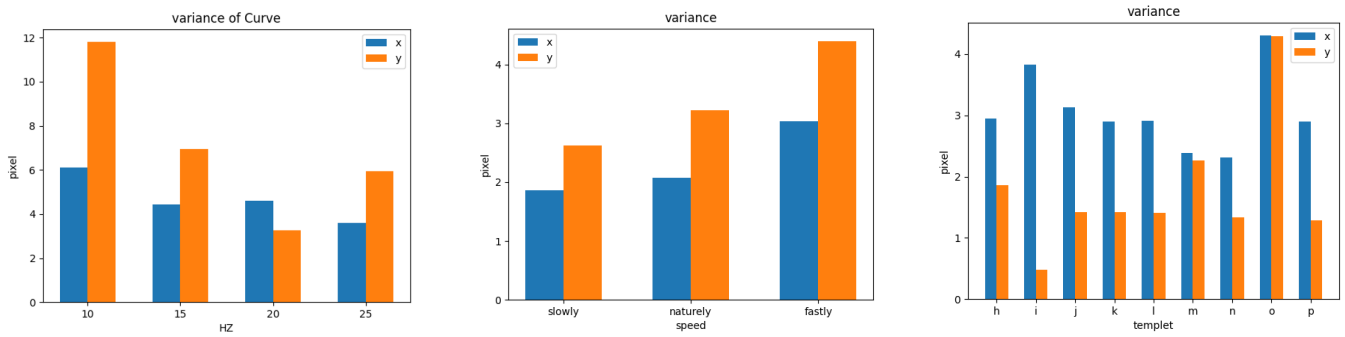
The influence of writing speeds——We test the drawing accuracy of a template 'Z' with three different writing speeds. We asked a participant to draw the template at three different speeds. The measured writing speed is 38cm/s, 19cm/s, and 9cm/s for the fast, nature, and slow speed, respectively. For each speed, we collect 10 samples to calculate the standard deviation. And the tracking accuracy at three different speeds is shown in Figure 9(b). We observe that as the speed increases, the accuracy gradually decreases.

The influence of different templets——We test the drawing accuracy of different templates. In the beginning, We show the participant nine different templates from 'h' to 'o'. And the participant draws the nine templates at 9cm/s. For each templet, we also collect 10 samples. The drawing accuracy of different templets are shown in Figure 9(c). Observe the figure, there is no significant accuracy difference at different templets.

C. Text Writing

In this section, we limit the application scenario to classroom teaching, and the experimental data only collects commonly used letters. Participants are required to draw the corresponding letters according to the given pictures, and the writing speed is kept as constant as possible during the writing process. Each letter drew 11 times, stitch 10 samples into a training set, and leave one for as a test set. Here, we print several writing examples, including 26 letters, 2 numbers, a triangle, and a Chinese character "you".

recognition accuracy——We asked participants to write 26 letters, each letter ten times. Then other participants



(a) different sampling rates (b) different writing speeds (c) different templets

Fig. 9. the tracking accuracy under various settings

identify them. The recognition accuracy is shown in Table I. The letter 'g' and the letter 'h' are indistinguishable from the letter 'y' and the letter 'n' due to errors. The letter 'o' unrecognizable because it cannot be connected end to end.

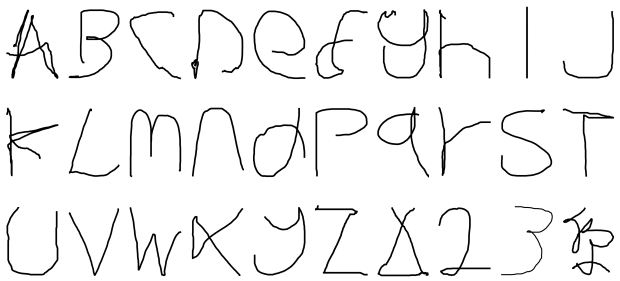


Fig. 10. Writing sample, including 26 letters, 2 numbers, a triangle, and a Chinese character "you",

TABLE I
recognition accuracy

	A	B	C	D	e	f	g	h	I
perc	100	100	80	80	90	60	20	60	100
	J	k	l	m	n	o	p	q	r
perc	70	50	90	80	100	10	70	100	100
	s	T	U	v	w	x	y	z	
perc	90	90	100	90	80	100	60	100	

VI. LIMITATION AND DISCUSSION

In this part we present and discuss some limitations of this work.

A. Tracking Accuracy

Accumulating the displacement between each frame through a machine learning pipeline depends on the dataset excessively. In a specific application scenario, due to the different writing habits of each person, varying degrees of error always exists even in the writing of the same person, which leads to the same writing objects correspond to different acceleration sequences. Hence, there is a difference between the calculated displacement values and the template both in x and y directions. Besides, the pipeline model cannot calculate the abrupt value. At the same time, there exist misalignment between the calculated value and the template value. This misalignment caused by a data synchronization error will cause a delay or error during the application. Besides, different writing objects correspond to different acceleration sequences, so the system cannot identify writing objects that have not been collected. In future work, we hope to synchronize data streams through synchronization signals to avoid errors caused by manual synchronization. and collect more datasets.

B. Error Accumulation

Compared with a double integral calculation of displacement, our system avoids the influence of calculation error in the previous time on the displacement of the next time. However, there are still errors between the displacement calculated by the pipeline model and the true displacement. After all segments of the displacement are accumulated to form a trajectory, these errors will be accumulated to a large cumulative error. Observe the triangle sample in Figure 10, the triangle drawn by the system is different from the template figure. Different from the finger tracking system based on computer vision and magnetic field, which use the fingerprint algorithm to directly locate the position of the object frame by frame to form a trajectory, our system cannot completely avoid the error accumulation by accumulating the displacement. It can only be improved by improving accuracy. That is determined by the inherent characteristics of inertial sensors.

Finally, due to the limitations of experimental equipment and the number of participants, we can not conduct detailed experiments on influencing factors here. At the same time, because of the patience of machine learning data, the dataset collected by experimenters with very different writing habits have large recognition errors. It

is worth noting that the pipeline model cannot correctly calculate the corresponding displacement if there is a large speed deviation between training data and test data.

VII. CONCLUSION

In this work, we implemented a fine-grained finger tracking system based on inertial sensors. Use acceleration sequences, our system can draw the corresponding trajectory. The principle is to regard the acceleration of the first 2.4 seconds (the sampling rate is 25HZ, 60 in total) as the feature of displacement. This method avoids the accumulative errors caused by the double integral calculation of the displacement. Experiments show the error is less than 2.32mm per frame under the 25HZ sampling rate and draw speed is less than 30cm per second.

Few studies use inertial sensors for continuous fine-grained finger tracking but discrete motion classification. In this work, we try to use inertial sensors to perform fine-grained continuous tracking in a new way, hoping to explore different ways of using inertial sensors. In future work, we hope to choose a more appropriate model. We will also explore the combination of this new application method and other sense methods.

References

[1] Daniel Ashbrook, Patrick Baudisch, and Sean White. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, page 2043–2046. Association for Computing Machinery, 2011.

[2] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion Capture of Hands in Action Using Discriminative Salient Points, volume 7577. 2012.

[3] S. Bovet, H. G. Debarba, B. Herbelin, E. Molla, and R. Boulic. The critical role of self-contact for embodiment in virtual reality. IEEE Transactions on Visualization and Computer Graphics, 24(4):1428–1436, 2018.

[4] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. utrack: 3d input using two magnetic sensors. In Proceedings of the 26th annual ACM symposium on User interface software and technology, page 237–244. Association for Computing Machinery, 2013.

[5] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. Finexus: Tracking precise motions of multiple fingertips using magnetic sensing. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, page 1504–1514. Association for Computing Machinery, 2016.

[6] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 12(null):2121–2159, 2011.

[7] Chris Harrison and Scott E. Hudson. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In Proceedings of the 22nd annual ACM symposium on User interface software and technology, page 121–124. Association for Computing Machinery, 2009.

[8] Chris Harrison, Desney Tan, and Dan Morris. Skinput: appropriating the body as an input surface. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, page 453–462. Association for Computing Machinery, 2010.

[9] Fang Hu, Peng He, Songlin Xu, Yin Li, and Cheng Zhang. Fingertrak: Continuous 3d hand pose tracking by deep learning hand silhouettes captured by miniature thermal cameras on wrist. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 4(2):Article 71, 2020.

[10] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. Digitspace: Designing thumb-to-fingers touch interfaces for one-handed and eyes-free interactions. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, page 1526–1537. Association for Computing Machinery, 2016.

[11] W. Kim, J. Song, and F. C. Park. Closed-form position and orientation estimation for a three-axis electromagnetic tracking system. IEEE Transactions on Industrial Electronics, 65(5):4331–4337, 2018.

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. International Conference on Learning Representations, 2014.

[13] Dong Li, Jialin Liu, Sunghoon Ivan Lee, and Jie Xiong. Fm-track: pushing the limits of contactless multi-target tracking using acoustic signals. In Proceedings of the 18th Conference on Embedded Networked Sensor Systems, page 150–163. Association for Computing Machinery, 2020.

[14] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. Wifinger: talk to your smart devices with finger-grained gesture. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, page 250–261. Association for Computing Machinery, 2016.

[15] J. Liu, G. Wang, P. Hu, L. Duan, and A. C. Kot. Global context-aware attention lstm networks for 3d action recognition. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3671–3680, 2017.

[16] Jess McIntosh, Paul Strohmeier, Jarrod Knibbe, Sebastian Boring, and Kasper Hornbæk. Magnetips: Combining fingertip tracking and haptic feedback for around-device interaction. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, page Paper 408. Association for Computing Machinery, 2019.

[17] Adiyan Mujibiya, Xiang Cao, Desney S. Tan, Dan Morris, Shwetak N. Patel, and Jun Rekimoto. The sound of touch: on-body touch and gesture sensing based on transdermal ultrasound propagation. In Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces, page 189–198. Association for Computing Machinery, 2013.

[18] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, page 1515–1525. Association for Computing Machinery, 2016.

[19] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. Auraring: Precise electromagnetic finger tracking. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 3(4):Article 150, 2019.

[20] Dario Pavllo, Thibault Porssut, Bruno Herbelin, and Ronan Boulic. Real-time finger tracking using active motion capture: a neural network approach robust to occlusions. In Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games, page Article 6. Association for Computing Machinery, 2018.

[21] S. Song, C. Hu, B. Li, X. Li, and M. Q. Meng. An electromagnetic localization and orientation method based on rotating magnetic dipole. IEEE Transactions on Magnetics, 49(3):1274–1277, 2013.

[22] S. Song, H. Ren, and H. Yu. An improved magnetic tracking method using rotating uniaxial coil with sparse points and closed form analytic solution. IEEE Sensors Journal, 14(10):3585–3592, 2014.

[23] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. ACM Trans. Graph., 35(4):Article 143, 2016.

[24] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans. Graph., 33(5):Article 169, 2014.

[25] Hoang Truong, Shuo Zhang, Ufuk Muncuk, Phuc Nguyen, Nam Bui, Anh Nguyen, Qin Lv, Kaushik Chowdhury, Thang Dinh, and Tam Vu. Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition. In Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, page 54–67. Association for Computing Machinery, 2018.

[26] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology, page 851–860. Association for Computing Machinery, 2016.

[27] Eric Whitmire, Farshid Salemi Parizi, and Shwetak Patel. Aura: Inside-out electromagnetic controller tracking. In Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, page 300–312. Association for Computing Machinery, 2019.

[28] Dan Wu, Ruiyang Gao, Youwei Zeng, Jinyi Liu, Leye Wang, Tao Gu, and Daqing Zhang. Fingerdraw: Sub-wavelength level finger motion tracking with wifi signals. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 4(1):Article 31, 2020.

[29] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. Opisthenar: Hand poses and finger tapping recognition by observing back of hand using embedded wrist camera. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, page 963–971. Association for Computing Machinery, 2019.

[30] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Ruichen Meng, Sumeet Jain, Yizeng Han, Xinyu Li, Kenneth Cunefare, Thomas Ploetz, Thad Starner, Omer Inan, and Gregory D. Abowd. Fingerping: Recognizing fine-grained hand poses using active acoustic on-body sensing. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, page Paper 437. Association for Computing Machinery, 2018.

[31] Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. Skintrack: Using the body as an electrical waveguide for continuous finger tracking on the skin. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, page 1491–1503. Association for Computing Machinery, 2016.