Headmotion: Human-Machine Interaction Requires Only Your Head Movement

Duoteng Xu[1(✉)], Peizhao Zhu[2], and Chuyu Zheng[3]

[1] College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518061, Guangdong, China
2019112144@email.szu.edu.cn
[2] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518061, Guangdong, China
[3] College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518061, Guangdong, China

Abstract. With increasing demand for smart wearable devices and the booming development of pervasive computing, more and more new human-computer interaction methods for wearable devices are proposed to make up for the shortcomings of traditional wearable device interaction methods and improve the efficiency and ubiquity of interaction.

Up to now, the human-computer interaction of wearable devices is still dominated by contact interactions, such as touching the screen and pressing physical buttons. This interaction method is convenient in most scenarios, but there are limitations in some situations, such as disabled people cannot use their hands for human-computer interaction, and drivers are not suitable for touch interaction with their hands when driving. To address this shortcoming, we designed a natural interaction system for ear-worn smart devices, which includes a deep network recognition model based on acceleration, angular velocity and angle data of head motion collected by inertial units worn on the ear. The interaction system now can realize human-computer interaction of head motion in various complex scenarios, which greatly liberates the user's hands. We have experimentally verified the excellent accuracy and real-time performance of our designed system.

Keywords: Human-computer interaction · IMU · Android · Machine learning

1 Introduction

With the rapid development of information technology, more and more studies have begun to focus on human behavior identification and human-computer interaction, and strive to improve people's living standards through science and technology. At present, the ways to realize human-computer interaction are traditional interaction and new interaction. Among them, traditional interaction methods, such as keyboard interaction, touch screen interaction, have been widely used in our lives. However, traditional interaction will bring some inconvenience to people, such as keyboard, touch screen interaction

K. Wu et al. (Eds.): ICECI 2021, LNICST 437, pp. 98–107, 2022.
https://doi.org/10.1007/978-3-031-04231-7_9

requires the use of hands and eyes, voice interaction must be used in a quiet environment and easy to disturb others. Under the inconvenience brought by the traditional way of interaction, the new way of interaction arises at the historic moment.

Some researchers have devoted themselves to the interactive intelligence of existing wearable devices. Hui-Shyong Yeo [1] and others designed a smartwatch WatchMI. Users realize various functions of smartwatch by touching, twisting and translating smartwatch, such as music player, clock setting, map display, text input, control of remote devices and so on. To some extent, this interaction method solves the problem of thick fingers on the watch screen, and it is also widely used, but it still needs to use both hands to complete the interaction, so it is inconvenient to use.

Some researchers are devoted to the study of physiological signals, such as muscle sound signals (MMG), ECG signals (ECG) and so on. Tianming Zhao [2] proposed a gesture recognition system based on photoelectric volume pulse wave detection (PPG), which can use commercial wearable devices to recognize fine-grained gestures at the finger level. The system detects the blood flow of the wrist, and when the hand moves, the blood flow changes, so as to realize the hand movement detection. This kind of interaction does not need to use the eye to gaze at the screen, uses the hand to complete the motion detection. Obviously, the method of detecting activity through physiological signals is more natural than "virtual devices" and 'smart devices". On the other hand, due to the particularity of physiological signals, it is difficult to guarantee the accuracy of the system.

The design and implementation of the HeadMotion natural interaction system for ear-wearing intelligent devices can detect the user's head movement in real time and interact naturally in the case of freeing hands. This kind of interaction enables users to interact at will, and is suitable for a variety of scenarios, which will bring great convenience to people in use. Compared with other interactive methods, it can solve the problem that two hands can not be used to control the device in special cases, because the user can control the device by recognizing the head movement. At the same time, this interaction mode does not have high requirements for hardware, and any earwear device or head-mounted device with inertial measurement unit (IMU) can realize the system.

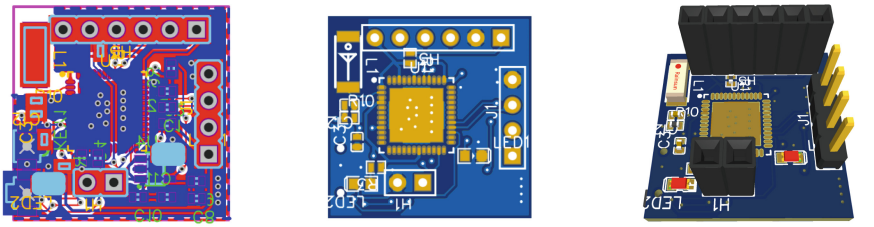
2 Hardware and Software Design of Equipment

2.1 Hardware Introduction

In order to make the collected data as close as possible to the actual product application scenario, we need to build the hardware platform of this system on the headset. However, the headphones sold in the market at this stage are highly integrated, which makes it difficult for us to carry out secondary development of software and hardware. To solve this problem, we independently developed a set of data acquisition devices that can support ear-worn. The design of hardware circuits and product shells take into account various factors such as versatility and resource consumption, and maximize the restoration of the user wearing the headset, making it easy to port to traditional headphones.

After several iterations of updates, the final version of the ear-worn device consists of a nine-axis IMU, a microcontroller ESP32, a 3.7 V Li-ion battery and 3D printed

parts, whose PCB layout and 3D model diagram are shown in Fig. 1, and the final physical drawing is shown in Fig. 2. When the user's head moves, the IMU above the ear is simultaneously driven and collects signals of angular velocity, angle and linear acceleration in the X, Y and Z directions in the 3D coordinate system and transmits them to the microcontroller ESP32 via serial communication, for a total of $3 \times 3 = 9$ channels. The microcontroller ESP32, as the master chip, receives the signals measured by the IMU and transmits the data wirelessly to the Android application on the cell phone using the Bluetooth function that comes with the ESP32. In addition, the hardware platform is powered by a 3.7 v miniature lithium battery.



（a）diagram of printed circuit board　（b）2D model of printed circuit board　（c）3D model of printed circuit board

Fig. 1. Printed circuit board of hardware system

（a）Housing 3-D model of printed circuit board　（b）Physical picture of hardware

Fig. 2. Housing 3-D model of printed circuit board and Physical picture of hardware system

2.2 Hardware Programming

The embedded software is based on the Arduino platform, and is programmed on Arduino IDE using Champact + language. With the help of the rich open source function library in Arduino, it is convenient to complete the basic driver of each module, such as the library "JY901.h" which depends on IMU and the library "BluetoothSerial.h" of ESP32. UART reads IMU data, calculates Euler angle from quaternion, and Bluetooth sends head action data.

The nine-axis IMU JY901 we use includes an accelerometer, a gyroscope and a magnetometer, which can obtain the linear acceleration, angular velocity and magnetic

field of the user's head movement. Because the data obtained by the gyroscope has white noise when measuring the angular velocity, the method of using the angular velocity to obtain the Euler angle directly will bring the integration error. Better results can be achieved by fusing all the original motion data through a stable filter. Therefore, we add Kalman filter to the hardware data acquisition program to obtain more accurate and stable data.

For head movement, posture is also an important feature to distinguish different head movements. Therefore, it is very important to get posture from the original data. The JY901 module we use has an integrated attitude solver, which can calculate the motion attitude more quickly and represent and output it in quaternion format. For hardware devices, both quaternion and Euler angle can be used to describe attitude, but Euler angle is closer to people's physical understanding of attitude and is more conducive to our subsequent data feature analysis, so we use the following formula (1) to convert the quaternion output of JY901 module into Euler angle on the hardware side [3].

$$\begin{cases} pitch = \arcsin[2(q_0q_2 - q_1q_3)] \\ roll = \arctan \frac{2(q_0q_1+q_2q_3)}{q_0^2-q_1^2-q_2^2+q_3^2} \\ yaw = \arctan \frac{2(q_1q_2+q_0q_3)}{q_0^2+q_1^2-q_2^2-q_3^2} \end{cases} \tag{1}$$

where:

pitch—Pitch angle, range from −90° to 90°;
roll—Roll angle, range from −180° to 180;
yaw—Heading range from −180° to 180;
q_0, q_1, q_2, q_3—Quaternion.

3 Experimental Environment and Database Construction

This system requires subjects to wear our home-made hardware device to acquire inertial unit data during head movements to train the head movement classification model. For this purpose, we invited eight volunteers, aged between 20 and 24 years old, who were all students, to participate in the database building. Each subject completed a specified set of movements wearing our home-made hardware device, including twelve movements: head down, head down twice, head up, head up twice, head left lean, head right lean, head left turn, head right turn, head left turn twice, head right turn twice, head left turn then right turn, head right turn then left turn, and each movement was repeated 50 times. The data was numbered from P1 to P8 according to the subjects. In order to facilitate the presentation of the results, the twelve movements were numbered, and the corresponding numbers of each movement are shown in Fig. 3.

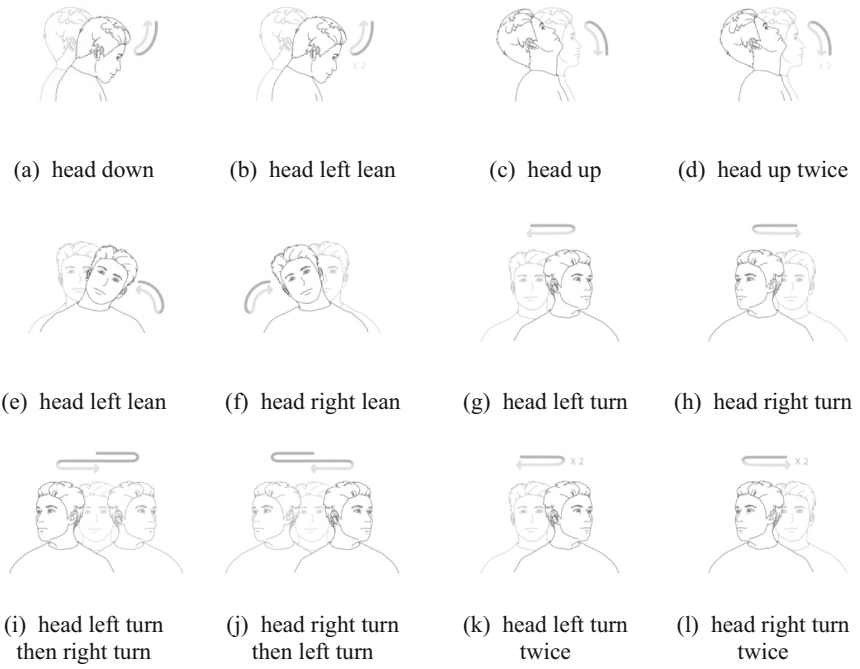


Fig. 3. Schematic diagram of head movement

4 Construction of the Human-Computer Interactive System

4.1 Overall Architecture of Interactive System

Figure 4 shows the overall block diagram of the system. The system is mainly divided into head posture data acquisition, data transmission, data visualization and preservation, model training, motion detection/segmentation, motion recognition, natural interaction and so on. Here we introduce the whole system through the training process and application process.

HeadMotion Training Process. Users wear ear-wearing smart devices for head movements. After the head movements are sensed by intelligent devices, the main control chip ESP32 reads the signals collected by IMU through UART, and then transmits the IMU signals to the mobile phone through its own Bluetooth controller. After receiving the data sent by Bluetooth, the Android application software on the mobile phone visualizes and saves the data, and establishes the data set needed for model training in the database. The PC side reads the data set, carries on the data preprocessing, the feature engineering, the training model, then derives the machine learning model, and finally deploys the machine learning model to the Android application software.

HeadMotion Application Process. When the user actually uses the earpiece smart device, the posture data flows to the mobile phone through the same process. At this

time, the mobile phone can detect/segment the head movement in real time and recognize the head movement. The mobile phone will control the Map application to perform corresponding operations (such as map view movement and zoom, etc.) according to different head movements, and the response results of the Map application will be fed back to the user in real time, thus completing the whole process of natural interaction.

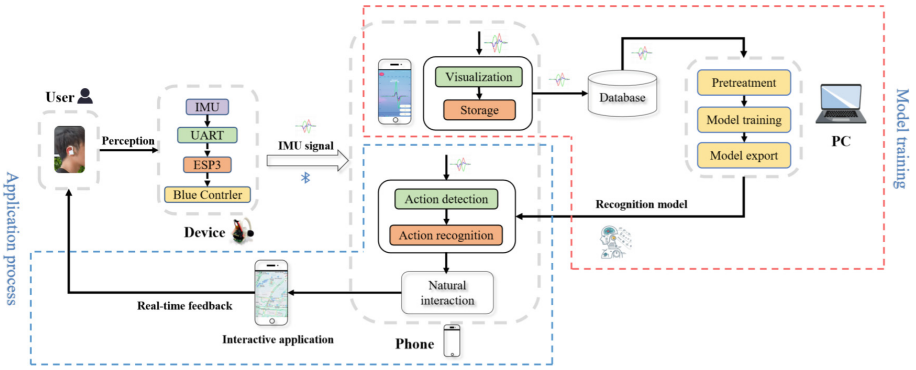


Fig. 4. The overall block diagram of the system

4.2 Construction of Recognition Model

The traditional convolution neural network has good performance in extracting the features of single frame signal, but it is poor in learning time series with a long time span. Yoshua Bengio pointed out in the reference that with the continuation of the time step, the current error messages will not affect the iterations with a long time span [4]. In view of this, Hochreiter and Schmidhuber proposed a special network, long-term and short-term memory network (Long short-term memory, LSTMs) [5], which introduces memory gate and forget gate on the basis of the traditional cyclic neural network, and determines when to remember or ignore the input in the hidden state through the special mechanism, so as to control the span of the cyclic neural network in the time-dependent relationship.

In order to learn the dependence of the time series data of head movements in the time dimension, so as to better classify the head movements, we choose the long short-term memory neural network to model the collected data. Considering the power consumption and efficiency, we finally choose the three-axis angular velocity time series data as the input of the network, which can achieve a very good classification effect.

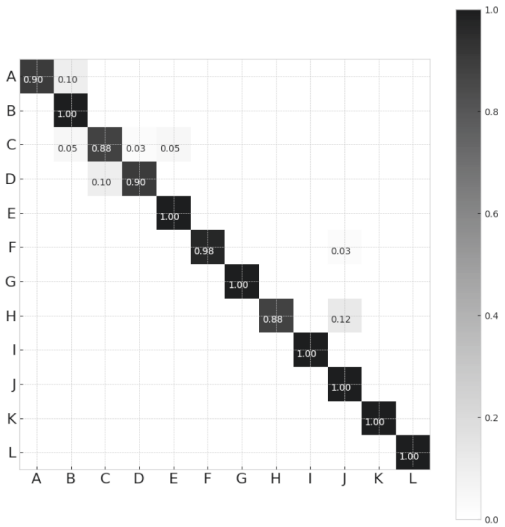
4.3 Evaluation of Recognition Model

We conducted two sets of experiments on the data of 8 people collected.

Non-cross-Person Experiment. In this experiment, the data of each subject is trained and tested separately. Thanks to the strong learning ability of the LSTM network, it can achieve excellent performance in non-human experiments, and the classification results of non-human experiments are very excellent, basically reaching 100% classification accuracy.

Cross-Person Experiment. We performed a leave-one-out test on the data of 8 people collected. We randomly selected the data of one subject as the test set, and the remaining 7 bits as the training set for the experiment. The cross-human experimental test results are shown in Fig. 5. From the confusion matrix, we can see that the model performs well on the test set. Action A and action B, action H and action J repeat the same action again and again, so they already have a certain feature similarity in the original data, which is difficult to distinguish time periods for the endpoint detection model. Consequently, the two pairs of behaviors are easy to be confused in classification. Action C is easily mistaken for actions B, D and E. We suspect that this is because different volunteers have different movement habits, which will lead to personal characteristics of the data characteristics of head steering movements. It is comparatively easy to be confused in cross-person experiments. Except for the low recognition accuracy of action C (Head Lean Right), action D (Head Left Turn) and action H (Right Then Left), other head movements have been classified accurately.

Further modeling work is under way. Under the premise of controlling power consumption, we will further enhance the diversity of the input data of the identification model by merging the collected inertial group data and the calculated pose data.



'Down': 'A', 'Down_Twice': 'B', 'Lean_Left': 'C', 'Lean_Right': 'D','Left': 'E','Left_Then_Right': 'F', 'Left_Twice': 'G', 'Right': 'H','Right_Then_Left': 'I','Right_Twice': 'J', 'Up': 'K', 'Up_Twice': 'L'

Fig. 5. Confusion matrix for twelve actions

4.4 Endpoint Detection in Real-Time Interactive System

Endpoint detection, also known as voice activity detection, is often used to intercept speech and non-speech segments in speech signals. In this system, in order to detect the head movement in real time, it is necessary to segment the data of each head action from the continuous data stream, so the endpoint detection technology in speech signal processing is applied to this system. Common endpoint detection methods include standard deviation detection, constant false alarm (CFAR) detection and so on.

- **Standard deviation detection.** The methods detect the beginning and end of the signal according to the real-time standard deviation of the signal. This method can achieve a good detection effect by adjusting the appropriate threshold in the case of constant noise [6]. However, it is not suitable to apply this method when the noise is unclear or the noise will change.
- **CFAR.** The methods can obtain noise according to the environment and obtain the threshold dynamically according to the noise, so as to detect the signal. The algorithm performs well under the condition of stable noise and is often used for signal detection in atmospheric noise and clutter [7]. The signal noise of this system comes from the signal with high sensitivity such as angular velocity. The sudden change caused by this kind of signal will affect the performance of CFAR, and the noise threshold is too large to detect the head movement.

This system adopts speech activity detection based on statistical model proposed by J. Sohn, N. S. Kim [8]. In this method, a robust speech activity detector for variable rate speech coding is developed, and the developed speech activity detector uses decision-oriented parameter estimation method for likelihood ratio test. In this method, the short-time Fourier transform (STFT) of the signal is calculated, and the noise spectrum is estimated from the noise using the minimum statistics, thus the endpoint detection is carried out. The performance of this system is better than the traditional methods such as standard deviation detection, peak detection, constant false early warning and so on.

5 Actual Interactive Application

Head movement-based interaction system to meet the needs of achieving various scenarios of interaction. We developed a companion Android application that implements the head motion contactless control of the map interface to verify that the designed interaction system is deployable and effective. The complete interaction system including the Android application is illustrated in Fig. 4, among which Baidu Map Android SDK is a set of application programming interfaces based on Android 4.0 and above devices. Developers can utilize the SDK to develop map applications for Android mobile devices. By calling the map SDK interface, developers can easily access Baidu Map services and data to build feature-rich and interactive map applications. The system integrates Baidu Map into Android application software. After the system detects and recognizes the user's head movements in real time, the application software will use the application program interface provided by Baidu Map to control the map instead of the user's hands,

such as map view movement and zooming, to complete the process of natural interaction according to the head movement recognition result.

The correspondence between the user's head movements and Map application is shown in Fig. 6. When the system detects/recognizes the user's head movement, the application software simulates the user to slide the screen or click the zoom button to realize the change of map view according to the correspondence. Since the interaction process is dynamic, it is not convenient to display it statically, so only some results of the Map application view movement and zooming are shown in Fig. 7.

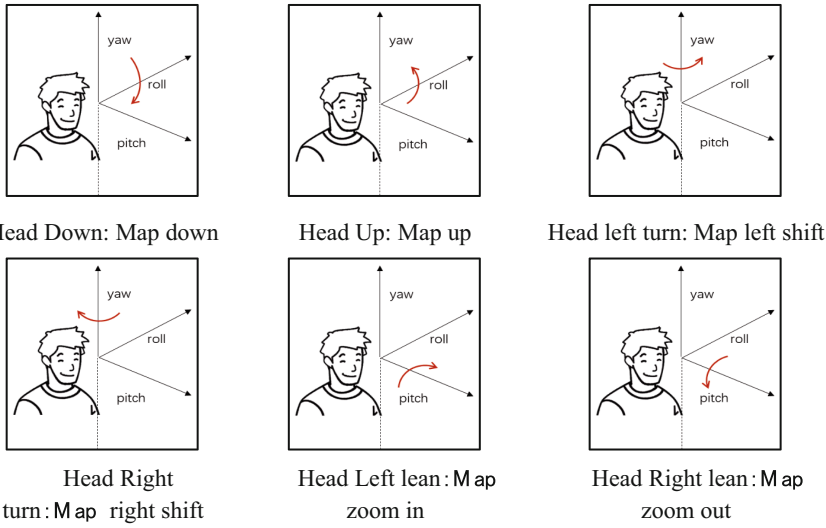


Fig. 6. Correspondence between head movement and Map Android application

Fig. 7. Moving and zooming of the perspective of the Map Android application

6 Summary and Prospect

Under the background of the popularity of wireless headphones, artificial intelligence headphones will be the next development trend of headphones. HeadMotion's smart headwear device has the advantages of simple hardware, high integration and low cost, and can be easily integrated into headset products. The machine learning model deployed on the application software is simpler and faster than the deep learning model, and the accuracy in the real-time system reaches 87.3%, which meets the actual needs of users. In addition, the corresponding data training model can be collected according to the needs of users, so as to customize the action categories and achieve the customization effect. HeadMotion interacts naturally with application software, and developers can easily add or modify the corresponding interactive functions. For example, HeadMotion can help users control the movement and zoom of the view of the map with head posture, the start and pause of the music player, and the answer and hang-up of the phone when driving or cycling, making the driving and cycling process more safe and convenient. Not only that, HeadMotion can also help people with disabilities to use smart wheelchairs, so that users can control the movement of smart wheelchairs with head posture and reduce the burden on their hands. HeadMotion can be used in any scenario where two hands cannot be used or are not convenient to use. All in all, HeadMotion really enables natural interaction that frees users' hands.

References

1. Yeo, H.S., Lee, J., Bianchi, A., Quigley, A.: WatchMI: pressure touch, twist and pan gesture input on unmodified smartwatches. In: International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI), Florence, pp. 394–399 (2016)
2. Zhao, T., Liu, J., Wang, Y., Liu, H., Chen, Y.: PPG-based finger-level gesture recognition leveraging wearables. In: IEEE International Conference on Computer Communications (INFOCOM), Honolulu, pp. 1457–1465 (2018)
3. Zhang, F., Cao, X., Zou, J.: A new conversion algorithm between full-angle quaternion and Euler angle. J. Nanjing Univ. Sci. Technol. **26**(4) (2002)
4. Liang, J., Chai, Y., Yuan, H., et al.: Emotion analysis based on polarity transfer and LSTM recursive network. J. Chin. Inf. **29**(5), 152, 159 (2015)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
6. Yi, S., Qin, Z., Novak, E., Yin, Y., Li, Q.: GlassGesture: exploring head gesture interface of smart glasses. In: IEEE International Conference on Computer Communications (INFOCOM), San Francisco, pp. 1–9 (2016)
7. Yu, T., Jin, H., Nahrstedt, K.: WritingHacker: audio based eavesdropping of handwriting via mobile devices. In: Proceedings of the ACM International Conference on Ubiquitous Computing (UbiComp), Heidelberg, pp. 463–473 (2016)
8. Sohn, J., Kim, N.S., Sung, W.: A statistical model-based voice activity detection. IEEE Signal Process. Lett. **6**(1), 1–3 (1999)