www.cadanda.com
Computer-Aided Design™
AND APPLICATIONS

Design and Implementation of Human-Computer Interaction Intelligent System Based on Speech Control

Jichao Liu[1], Wenhui Chang[1], Jing Li[1] and Ju Wang[1]

[1]Yanching Institute of Technology, Hebei Yanjiao, 065201, China, lijing@yit.edu.cn

Corresponding author: Jing Li, lijing@yit.edu.cn

Abstract. Introducing the voice control technology into the human-computer interaction of the mobile terminal can inject new vitality into the terminal device, simplify the operation difficulty, provide a brand-new experience for the user, and have a great market prospect. This paper discusses the sampling of speech signal and the method of frame windowing. Based on this, the method of extracting the characteristic coefficient of speech signal and its key parameters are discussed, including the concepts of short-term energy, power, zero-crossing mean in time domain and LPCC and MFCC parameters in the frequency domain. The DTW and HMM algorithms of speech recognition are studied. After describing their respective characteristics and solving methods, they are selected. Meanwhile, an embedded system software architecture was built. In order to verify the feasibility of the control system, the voice program was verified, including endpoint detection and DTW algorithm program. The verification results show that the voice control method of this paper is accurate and efficient for the control of human-computer interaction intelligent system.

Keywords: voice control; human-computer interaction; DTW; intelligent terminal.
DOI: https://doi.org/10.14733/cadaps.2020.S2.22-34

1 INTRODUCTION

With the rapid development of information technology in the era, the wave of intelligence has swept the world. People's work, study and life have become more and more closely linked with intelligent products, and the requirements for smart terminal equipment are getting higher and higher, especially handheld terminal Larson C R et al [1,2]. Since the size of the handheld portable terminal device is small, the available control devices (such as a keyboard) are not as much as ordinary PCs, so it is particularly important to design the control mode of the terminal [3]. A well-designed terminal control system should have the characteristics of simple operation, easy to get started, and complete and rich command functions Rogowski et al [4]. This creates a contradiction with the compact interface of the terminal device. In order to solve this contradiction, it is necessary to get rid of the constraints of physical control, and transfer to a more high-end control mode, voice recognition control is one of the most advantageous solutions Coelho Y L, Salomao J M et al [5,6].

Speech recognition technology is very rich in theoretical construction and practical application. Developed by AT&T Bell Labs, the Audrey speech discrimination system is the world's first computer-based speech recognition system [7]. This system can judge 10 English numbers based on voice, which has become the cornerstone of speech recognition technology research [8]. With the continuous maturity of computer science and technology, and also the advancement of speech recognition technology, scholars have scientifically dealt with various model problems caused by people's speech by constructing and applying linear prediction profiling techniques and dynamic regularization ideas D. Boucha et al[9,10]. Both have brought new ideas to the development of speech recognition. Some scholars have solved the problem of the establishment of the speech template library by using the academic theory of signal information estimation and decoding [11,12]. In view of the problem that the length of the sound emitted by the template matching in speech recognition is not uniform, the relevant scholars have constructed a dynamic time warping (DTW) technical framework to solve the problem [13,14]. In addition to the dynamic time warping algorithm, vector quantization (VQ) technology and hidden Markov model (HMM) were also proposed during this period, which brought a richer operation scheme to speech recognition. Some researchers have combined artificial neural network technology (ANN) with hidden Markov algorithm model, and applied to the development of long sentence continuous word discrimination system Rusko M et al [15,16]. Some scholars have proposed a normalized scoring mechanism that solves the problem of non-uniform speech duration [17,18]. This scoring mechanism greatly reduces the influence of speech duration on recognition scores. Some scholars have introduced linear predictive coding into speech recognition, which reduces the impact of speech frequency response changes on recognition [19]. Some studies have found that deep learning is applied to speech recognition, which is more than 20% better than the traditional GMM-HMM speech recognition system Yazdani R, Segura A et al [20]. Some studies have shown that acoustic models based on convolutional neural networks can achieve better performance than DNNs, achieving relatively 3%-5% performance improvement in large vocabulary continuous speech recognition tasks [21]. By using a very deep CNN structure, better recognition results can be obtained on some speech recognition tasks. In order to better model the dynamic correlation of speech timing, many researchers currently use the recursive structure based on long and short memory cells to model acoustic models and succeed in some large databases Pak C Let al [22]. However, the LSTM training is time-consuming and unstable. In order to make the feedforward neural network have the advantage of LSTM in temporal dynamic correlation, the delay neural network is a feedforward neural network that can be used for long-order dynamic correlation modeling. Re-applied to speech recognition, it has achieved good modeling effects in long-term context speech recognition tasks. However, the more mature deep learning-based acoustic modeling technology still relies on the traditional GMM-HMM for training, which leads to the training of the whole model into multiple stages. In order to solve this problem, how to carry out end-to-end speech recognition control has become a hot spot of research.

This paper implements a human-computer interaction intelligent terminal system with voice control. The instruction is a basic instruction, which is an identification of isolated words in the category of phonetics. Therefore, the choice to use the DTW algorithm allows for a better compromise between effectiveness and reduced system resource consumption. Therefore, the language programs in this paper are designed around the DTW algorithm. By researching the software at the bottom of the system, according to the needs of speech recognition, the overall software architecture of the system is established, and the device driver part is studied. Finally, the simulation of the written speech program is carried out, and the important parameter settings and methods of the program are discussed. The endpoint detection algorithm is optimized.

The rest of this article is organized as follows. Section 2 discusses the design of the voice control module, followed by the construction of the system software platform in Section 3 and the key driver design of the voice control intelligent terminal. Section 4 carries out simulation verification of the voice control human-computer interaction intelligent system. Section 5 summarizes the full paper and gives the future research directions.

2 DESIGN OF VOICE CONTROL MODULE

2.1 Speech Recognition System Architecture

The system for identifying sound mainly consists of the following components: a preprocessing module, a characteristic parameter extraction module, and an identification module, as shown in Figure 1.

The task of the endpoint detection module is to divide the starting point and the ending point for the speech segment to be identified; the task of the characteristic parameter extraction module is to convert the sampled speech into the frequency domain or the frequency domain or the scrambling domain, and save the result vector. First, the endpoint detection module detects the start and end points of the speech segment and completes the framing, and then the characteristic parameter extraction unit performs the calculation of the specified algorithm frame by frame, and then saves the result as the characteristic vector. For the isolated word authentication system, due to the scarcity of embedded system hardware resources, it is necessary to strictly control the operation and storage load of the system, which requires vector quantization of the feature vector. Of course, for a large-scale resource-rich form, the impact of these restrictions is negligible, and the result of the feature parameter extraction module is usually directly used as a speech recognition template. Then, the recognition algorithm unit performs a pairing calculation on the vector of the speech segment to be recognized and the vector of the template library, and sets the speech segment template with the largest degree of matching as the discrimination result output. Figure 2 shows the embedded system modules and architecture.

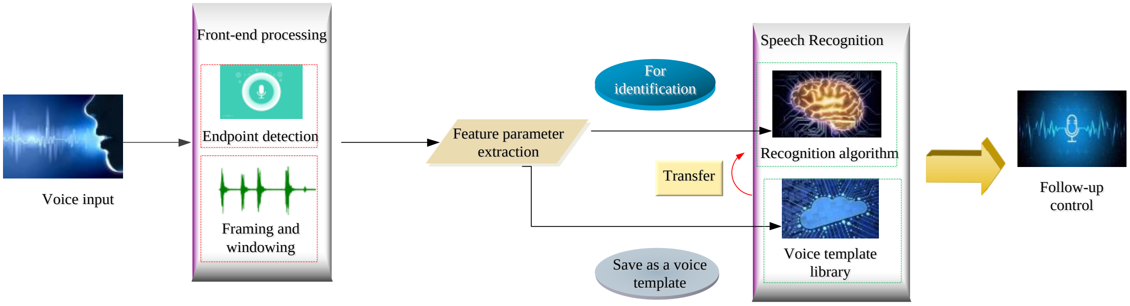


Figure 1: Speech recognition system block diagram.

2.2 Voice Signal Front-end Processing

1) Signal pre-strengthening

The speech signal will have a large loss in the high-frequency part, so it is necessary to make up for this part of the lost information by a means, so that the pre-emphasis processing comes into being. After the pre-emphasis process, the loss of the high-frequency part is recovered, and the signal quality of the entire spectrum is consistent, and can be processed in the same way. In addition, after pre-emphasis, the speech signal to be processed is only related to the utterance part, which is more in line with the conditions of frequency domain analysis and sound generation coefficient analysis.

Pre-emphasis can be achieved by processing the speech signal with the transfer function shown below:

$$G(z) = 1 - az^{-1} \tag{1}$$

The interval of a is [0.4, 1.0]. After processing the signal, the signal must be restored. The effect of the pre-emphasis function is removed.

2) Frame division and window function processing

The next step after the speech signal has been reinforced is to perform a "cut" process, using a window function to process the signal into speech micro-elements. Each of the micro-elements thus obtained is called a speech frame. Adjacent two micro-elements can be connected or overlapped. We generally use the latter because it can make the micro-elements too smooth. In order to make the interface between the frame and the frame good and the connection is stable, we generally adopt the overlapping framing method. The common area between the two frames is called frame shift, and the ratio of the frame shift to the length of the micro-element is [0, 0.5].

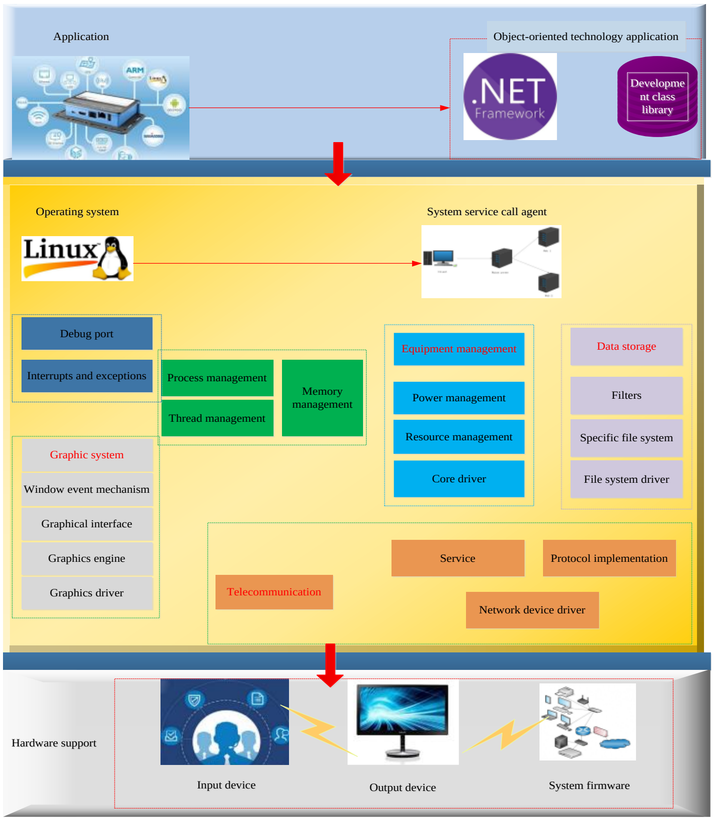


Figure 2: Embedded System Modules and Architecture.

Each 240 sampling points is set to one micro-frame, and 80 sampling points are set to the frame shift length, the ratio of which is 1:3. The method of dividing the frame is to integrate the speech signal V(n) with the framed window w(n) to obtain a "window processing" language signal function:

$$V_w(n)=V(n)*w(n) \tag{2}$$

Among the speech recognition processing, the most widely used window functions are rectangular window function, Hamming window function and Hanning window function. The expressions are:

$$w(n)=\begin{cases}1 & 0\le n\le K-1\\ 0 & other\end{cases} \tag{3}$$

$$w(n)=\begin{cases}0.54\text{-}0.46\cos(2\pi n/(K-1)) & 0\le n\le K-1\\ 0 & other\end{cases} \tag{4}$$

$$w(n)=\begin{cases}0.5[1-\cos(2\pi n/K)] & 0\le n\le K-1\\ 0 & other\end{cases} \tag{5}$$

K denotes the number of frames contained in each window, and the selection of w(n) is critical for the subsequent processing of the feature parameter extraction of the speech signal. Aiming at different application analysis environments and adopting a reasonable window model, the extracted feature parameters can be well matched with the actual language function waveform characteristics.

Comparing the above three kinds of window functions can be obtained: when the rectangular signal is used to frame the speech signal, the result will be the midpoint of the true frequency of the signal and the shape of the rectangular window spectrum. Spreading out, there is a phenomenon similar to "leakage", which is unfavorable for signal analysis. In addition, the low-pass waveform of the Hamming window is relatively flat, and the short-term nature of the signal can be described relatively completely, except that the resolution is not as high as the rectangular window function. Based on the above, the Hamming window can be used as the framing window.

3) Endpoint detection

The calculation steps of the time-based micro-energy zero-crossing mean two-threshold endpoint detection method are shown in Figure 3.

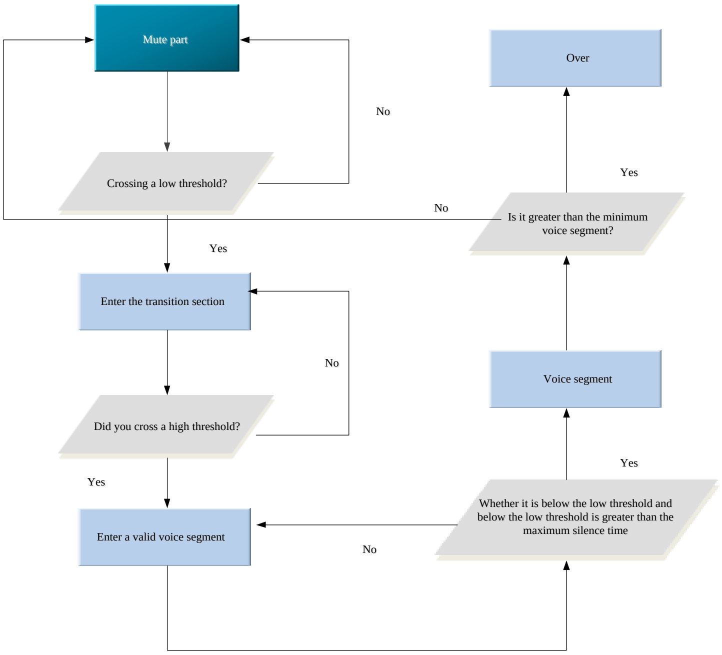

Mute part
Over
No
Yes
Crossing a low threshold?
No
Is it greater than the minimum voice segment?
Yes
Enter the transition section
No
Voice segment
Did you cross a high threshold?
Yes
Yes
Whether it is below the low threshold and below the low threshold is greater than the maximum silence time
Enter a valid voice segment
No

Figure 3: Flow chart of short-time energy short-time zero-crossing rate double threshold algorithm.

The so-called two-threshold endpoint detection method refers to the idea of using "double check", that is, time micro-element energy detection, time micro-element zero-crossing mean detection. The first step is to use the time micro-element energy to make the initial judgment, and the second step is to use the time micro-zero mean value to make the second recognition.

At the beginning of the detection process, four reference values are defined, which are the high and low thresholds of the energy of the time micro-element, and the high and low thresholds of the zero-crossing mean of the time micro-element. In this way, all speech detection processes are subdivided into four steps: a mute part, a transition part, a valid part, and a part. If the energy in the mute section, the zero-crossing mean, has crossed the low threshold, then the decision signal belongs to the transition. In the transition part, because the absolute value of the reference value is low, it is not yet possible to accurately answer whether it belongs to the effective sound part. Only when one of the above two reference values breaks through the high threshold value can the determination signal enter the effective sound part. If the reference value becomes a low threshold then we consider the signal to be at the end.

2.3 Speech Recognition Module Algorithm

1) The basic idea of the DTW algorithm
Dynamic time warping is a standard template matching comparison algorithm. This algorithm transplants the dynamic programming theory in operations research into sound recognition, and at the same time overcomes the problem of different sizes when comparing arrays of different speech function characteristic coefficients. Small vocabulary is widely used in the field of recognition and has achieved good results. The sound resolution program designed by the DTW method based on dynamic time planning not only has excellent performance, but also saves a lot of system resources and has good real-time performance. It is the preferred method for small vocabulary systems.

According to the dynamic programming algorithm, the unknown vector is uniformly extended or shortened to be equal to the length of the reference template of the template library. In such a program, the time coordinates of the speech to be recognized are irregularly warped or curved so that the characteristic vector of the speech matches the reference vector.
2) Solution of dynamic time warping algorithm
Assume:

$$i(1)=j(1)=1, g(1,1)=2d(a_1,b_1) \tag{6}$$

$$Sum(i,j)=\begin{cases} 0 & (i,j)\in region \\ big & (i,j)\notin region \end{cases} \tag{7}$$

In the formula, we generally set the range of the region to two parallel vertices with (1,1), (i, j) and a parallelogram with a slope of 0.5 and 2.

Analogy for a small pair of distances:

$$Sum(i,j)=\min\left\{\begin{matrix} Sum(i-1,j)+d(a_{i-1},b_j)\cdot w_n(1) \\ Sum(i-1,j-1)+d(a_i,b_j)\cdot w_n(2) \\ Sum(i,j-1)+d(a_i,b_j)\cdot w_n(3) \end{matrix}\right\},(i=2,3,\ldots,J,(i,j)\notin region) \tag{8}$$

The weights are set to:

$$w_k(1)=w_k(3)=1, w_k(2)=2 \tag{9}$$

The number of strokes of the adjustment function varies with the vector length of the template and the speech to be tested. It can be verified that if we define the weight function reasonably, the following formula will hold:

$$\sum w_n = I+J=Const \tag{10}$$

This way you can launch the pairing weights as:

$$L=Sum(I,J)/(I+J) \tag{11}$$

According to the laws of phonetics, we lock the range of the paired search route in a parallelogram with two opposite vertices at (1,1), (i, j), and the slopes of the two adjacent sides are 0.5 and 2, as shown in Figure 4.

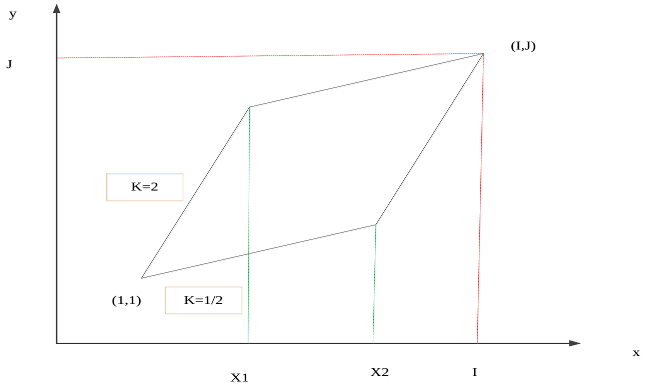


Figure 4: Schematic diagram of the DTW optimization algorithm.

It can be seen from Figure 4 that the pairing distance represented by the pairing points outside the restricted area is not required to be obtained, and in addition, it is not necessary to record the pairing path and the accumulated path array of all the frames. The reason for this is that during the calculation, we actually only used 3 possible matching points on the forward path. Grasping this point can greatly reduce the amount of DTW calculations and the necessary storage capacity of the system.

The accumulated route array can be expressed as:

$$D(x,y)=d(x,y)+\min[D(x-1,y),D(x-1,y-1),D(x-1,y-2)] \tag{12}$$

3 SYSTEM SOFTWARE PLATFORM CONSTRUCTION AND KEY CONTROL DESIGN OF VOICE CONTROL INTELLIGENT TERMINAL

3.1 Construction of the Software Platform

The system implementation mainly includes the following four levels: application, operating system, driver, and hardware device. The first step of the Linux system startup is the system homing of the central processor S3C2440 and the self-test of the structure. The second step is to perform the initial homing of the hardware devices related to the development board. The key point is to load the driver to the hardware.

In the operation of setting the kernel file, it is necessary to support the liquid crystal display, the support of the yaffs file format, the support of the UDA1341TS audio chip, the serial port and the support of the flash memory. After the setting is completed, we save the result of the setting as a file, and only need to load this setting file every time you want to call it.

The graphical user interface is also an important part of this article. In addition to the advantage of providing users with a friendly operator panel, the embedded QT system used in this article also has a development platform similar to VC: QT designer. This kind of tool brings great convenience to development. It can directly use the signal + slot mechanism that will be discussed later, and it is convenient to link functions and functions, functions and events.

The most distinctive feature of QT/Embedded is the programming of the signal + slot function. This mechanism connects the program event directly to the ideal response program, eliminating the difficulty of many program calls.

In this paper, QT is introduced in the speech recognition system. In the process of system startup, input of voice commands, output recognition results and control of the terminal, we have to present the results of the program to the user through the LCD.

When the voice program is written, we save it to the corresponding folder of the compiled QT. Finally, we compile this voice program in the QT directory. After successfully compiling, we use the command qvfb to realize the operation of the graphical user interface, then adjust some parameters, and finally run the voice program, you can see the visual voice program.

3.2 LCD Driver Design

The intelligent voice terminal uses the LCD as a display device, and the LCD can obtain strong support of the LINUX graphical user interface software, making the system design more user-friendly. The LCD driver directly serves the embedded QT. QT renders multiple pictures by calling various trace plotting functions in the driver.

The Linux operating system assigns a supported port to the graphical interface device. This port is the frame buffer device. This port virtualizes the buffer domain of the imaging device, ignoring the difference in hardware level of different types of imaging devices, and more advanced applications. The layer is capable of reading or writing data unobstructed in the imaging mode of operation to the imaging device buffer domain. Application programmers no longer have to study the specific orientation or save mode of the buffer domain of the underlying hardware layer imaging device, because these tasks have been done by the frame buffer startup program. From the perspective of the frame buffer component, by writing the color digitization value in the range of the image buffer domain to match the display, the corresponding color will appear on the LCD display. The organization structure of the frame buffer component driver is shown in Figure 5.

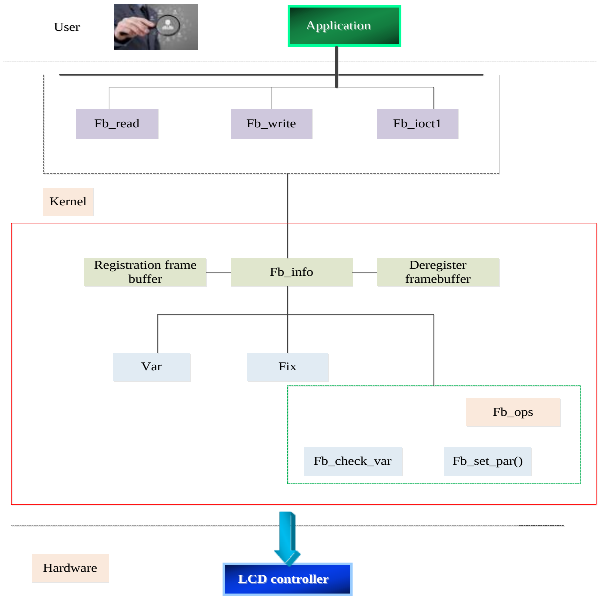


Figure 5: Organization of the frame buffer component driver.

3.3 Audio Chip UDA1341 Driver Design

The sound device startup program must operate the voice chip hardware in order to divert audio. After implementation, an audio mode port is connected to the application layer. Ports of the audio device face to the application layer, digital audio processing DSP and mixing device Mixer. The former function is to manage the transmission of the sound data stream, that is, the sound file in the digital format of the power amplifier, record the sound and save it as a digital format. The latter function is mainly for the sound release, that is, by mixing various sound frequencies. Controlling the sound quality is manifested by changing the loudness, frequency, etc. of the sound. The files describing these 2 ports are saved in the Linux device folder dev, the files dsp and mixer. The function of the audio driver we are designing mainly accomplishes the following tasks: the device returns to the default state, turns on the device, loads digital sound processing and mixing device drivers, and uninstalls the driver.

We create a UDA1341TS driver in the oss folder under the souud directory of the kernel file, mini2440_uda1341.C, then edit the code, and finally load the driver into these 2 files by modifying the settings reference files Makefile and Kconfig. When compiling the kernel, it will compile the written driver into the system. Once the audio device is turned on, the application can use the UDA1341 chip. When the program wants to change the sound size, the MADPLAY program will call the mixer in the driver to control the volume by modifying the function value.

4 VOICE-CONTROLLED HUMAN-COMPUTER INTERACTION INTELLIGENT SYSTEM SIMULATION VERIFICATION

4.1 Pre-processing Verification

Simulation based on Matlab programming are shown in Figure 6. The part of the two vertical lines is the effective voice part.

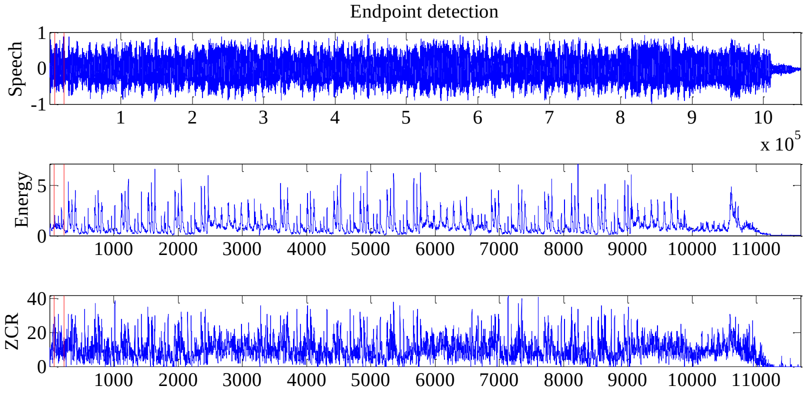


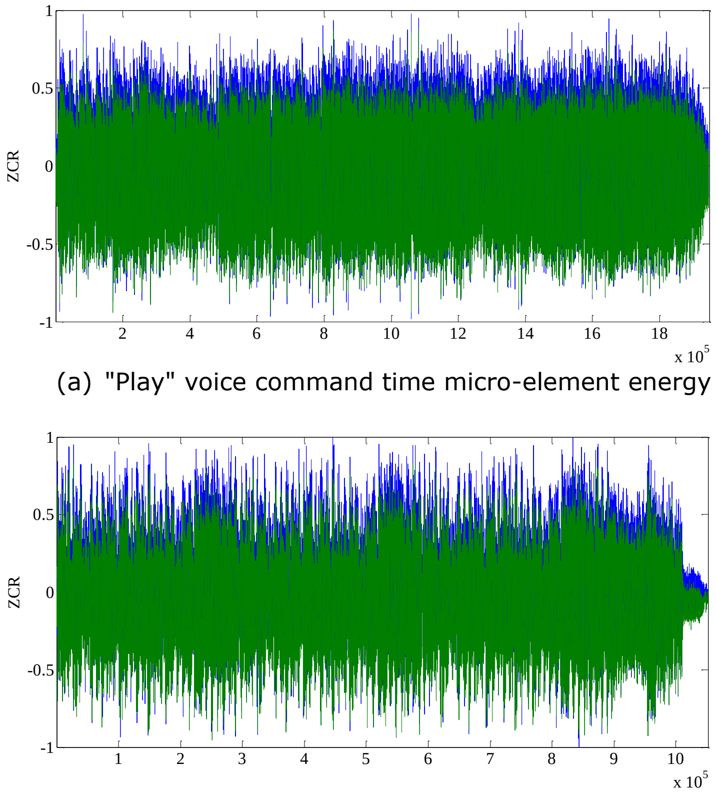
Figure 6: "Play" voice command endpoint detection simulation diagram.

It can be seen from Figure 6 that the effective speech part can be intercepted from a speech signal by the endpoint detection algorithm program, but the range of interception is still large, the precision is not high, and the latter optimization is needed. It need type endpoint detection algorithm for further processing.

In addition, based on Matlab, the concept of time micro-zero mean value and time micro-element energy value is simulated, and the time micro-element energy value and time-micro-zero mean value are obtained. The results are shown in Figure 7(a) and 7(b).

As can be seen from the above figure, in the effective speech segment, both the energy and the ZCR value peak, are respectively in the appropriate numerical interval, which indicates that the result detected by the endpoint is satisfactory.

Before performing endpoint detection, the speech signal should be used to do the framing operation of the add window function. The window function should be selected appropriately. The selected window length is too wide. When the calculation processing speed is increased, the recognition accuracy will decrease. Correspondingly, assuming that the window length is too narrow, although the accuracy of the system can be improved, the calculation time load of the system is increased. In view of the above situation, this topic uses the variable window function time micro-element endpoint detection method to apply a wider window function to the mute part. The narrower window function is applied to the transition part. The function is applied in the active segment.



(a) "Play" voice command time micro-element energy

(b) "Play" voice command time micro-element zero-cross mean

Figure 7: Time micro-element energy and micro-zero zero mean simulation.

The time micro-energy value has important significance for endpoint detection. It has many functions. Specifically: 1. The soft and turbid sounds contained in the speech signal have different bio-energy, and the unvoiced energy can be small and turbid. The sound energy is high, according to which it can identify the attribution of the speech segment; 2. On the basis of 1, the time micro-element energy can also help identify the valid speech segment and the silent segment in the algorithm; 3. In the Chinese character recognition, the time micro-element energy can distinguish between the initial and the final, and can effectively "cut" for long sentences. The following describes how to write the program:

(1) We declare the parameter int status, this variable can take 3 values, each value represents the stage of the speech segment, as follows: when status=0, indicating that the speech segment is in the silent position; when status=1, indicating the speech segment is in the transition part; when status=2, it indicates that the voice signal enters the valid part; when status=3, it indicates that the voice signal is in the end stage.

(2) When using the conventional endpoint detection method, the program should pre-process and set the state variable to zero. When the time micro-element energy exceeds the low threshold, the state variable is set to one. At this time, the voice becomes the transition zone, and then the wide window span is changed to a narrow span.

(3) When the time micro-element energy crosses the high threshold, the state variable is reset to 2, and the voice signal is declared to enter the effective area. At this time, the span of the window function is changed to an ordinary size.

(4) The time micro-element energy is gradually decreased. When it is less than the low threshold again, the window function span is changed again, and the process is completed when the speech signal is terminated.

In order to prevent the result of the error, the program also adds two variables, the narrowest speech width and the widest silent length. The improved Matlab simulation results of the speech signal endpoint detection calculated by the variable window function width method are shown in Figure 8.

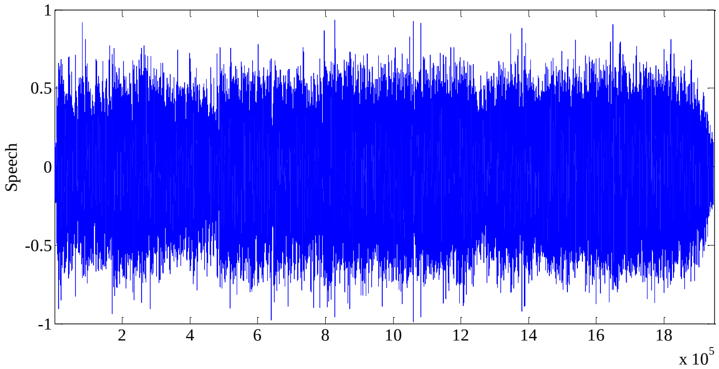


Figure 8: Variable window function width endpoint detection Matlab simulation.

The effective part of the speech segment is limited between the two vertical lines. Comparing this figure with the common endpoint detection algorithm, the reliability and accuracy of this method are higher, and the division interval is more reduced, thus saving a lot of system storage resources, eliminating a lot of redundant solution process, provide a good voice material for subsequent processing.

4.2 DTW Program Verification

We set up five control command samples, which are: "Play", "Pause", "Previous", "Next", "Stop". Based on these instructions, we performed ten sample recognitions of the Play command. The pairing path as shown in Table 1 was calculated by the Matlab simulation software.

	1	2	3	4	5	6	7	8	9	10
Play	3.32	3.21	3.01	3.03	2.97	2.88	2.66	3.56	3.41	3.01
Pause	8.99	8.56	8.99	7.98	9.32	9.86	9.66	8.55	9.20	8.97
Previous	8.45	8.11	8.51	8.36	8.63	9.63	9.32	9.24	9.44	9.11
Next	8.55	8.21	8.71	9.26	9.61	8.01	7.99	10.22	9.32	9.10
Stop	7.31	8.23	8.10	8.01	8.20	9.33	9.01	8.75	8.49	8.90

Table 1: Using Matlab to simulate the normal DTW method pairing difference.

After the above, this paper uses the DTW optimization algorithm to perform the sampling identification of ten “forward” instructions, and again uses the Matlab simulation software to obtain the pairing path results as shown in Table 2.

	1	2	3	4	5	6	7	8	9	10
Play	1.21	2.02	1.99	1.55	3.42	1.62	2.81	2.52	1.37	1.44
Pause	8.21	9.13	8.01	8.72	8.51	9.01	9.22	9.45	8.23	9.31
Previous	9.23	8.66	8.32	9.56	9.14	9.00	8.01	9.04	8.55	7.99
Next	8.53	8.32	9.88	7.99	7.98	9.71	8.02	8.47	8.65	8.75

Stop	9.59	7.81	9.13	8.73	8.73	8.49	9.85	8.26	8.99	8.37

Table 2: Using Matlab simulation improved DTW method pairing difference.

According to the above results, this paper averages the pairing difference of each group of instructions, and uses a certain range of this average value as the effective recognition range. When the error of the input command and the template is within this range, the system determines the input voice. The instruction corresponding to the template performs subsequent operations. Otherwise, the judgment is negative, and the sample is returned.

5 CONCLUSION

As an important way of human-computer interaction, voice control has the characteristics of interaction and convenience compared with other interaction methods, making human-computer interaction more natural and easy, liberating users' hands and improving the automation of the system, which will become the most natural way of human-computer interaction. In this paper, the method of speech recognition is studied in detail, and the front-end processing of signals, important parameters such as time-domain frequency domain and dynamic time warping algorithm are analyzed. On the basis of the above, a speech recognition program was written to recognize the speech and achieved good results. In order to apply the algorithm to practice, the embedded system platform was built and the visual GUI design was completed. The voice program was transplanted to QT, and the "play", "pause", "previous" and "last", The "stop" 5 basic control commands are implemented. However, the implementation of this article is carried out in a quiet and ideal environment. However, the actual application environment is very different, and the noise interference is very serious. How to "filter" effective speech information from it, we need to study anti-noise algorithms, such as speech enhancement algorithm spectral subtraction, Wiener filtering and compensation techniques. These also require more in-depth research.

ACKNOWLEDGEMENTS

Hebei Youth Fund Project: Development and Research of Pension Robot (No. QN2019328).

Jichao Liu, https://orcid.org/0000-0001-5778-5417
Wenhui Chang, https://orcid.org/0000-0001-5960-6988
Jing Li, https://orcid.org/0000-0003-3027-4355
Ju Wang, https://orcid.org/0000-0001-5928-0519

REFERENCES

[1] Larson, C. R.; Altman, K. W.; Liu, H. J., et al.: Interactions between auditory and somatosensory feedback for voice F0 control, Experimental Brain Research, 187(4), 2008, 613-621. https://doi.org/10.1007/s00221-008-1330-z

[2] Ballesteros, F.; Soriano, E.; Guardiola, G., et al.: The Plan B OS for ubiquitous computing. Voice control, security, and terminals as case studies, Pervasive and Mobile Computing, 2(4), 2006, 472-488. https://doi.org/10.1016/j.pmcj.2006.08.001.

[3] Fujiki, R. B.; Chapleau, A.; Sundarrajan, A., et al.: The Interaction of Surface Hydration and Vocal Loading on Voice Measures, Journal of Voice Official Journal of the Voice Foundation, 31(2), 2017, 211. https://doi.org/10.1016/j.jvoice.2016.07.005.

[4] Rogowski, A.: Web-based remote voice control of robotized cells, Robotics and Computer-Integrated Manufacturing, 29(4), 2013, 77-89. https://doi.org/10.1016/j.rcim.2012.11.002.

[5] Coelho, Y. L.; Salomao, J. M.; Kulitz, H, R.: Intelligent Hand Posture Recognition System Integrated to Process Control, IEEE Latin America Transactions, 15(6), 2017, 1144-1153. https://doi.org/10.1109/tla.2017.7932703.

[6] Vanus, J.; Smolon, M.; Koziorek, J., et al.: Voice Control of Technical Functions in Smart Home with KNX Technology, Lecture Notes in Electrical Engineering, 330, 2015, 455-462. https://doi.org/10.1007/978-3-662-45402-2_68.

[7] Liu Z. T.; Min W.; Cao W. H., et al.: A facial expression emotion recognition based human-robot interaction system, IEEE/CAA Journal of Automatica Sinica, 4(4), 2017, 668-676. https://doi.org/10.1109/JAS.2017.7510622.

[8] Long, Y.; Aleven, V.: Educational Game and Intelligent Tutoring System: A Classroom Study and Comparative Design Analysis, ACM Transactions on Computer-Human Interaction, 24(3), 2017, 1-27. https://doi.org/10.1145/3057889.

[9] Boucha, D.; Amiri, A.; Chogueur, D.: Controlling electronic devices remotely by voice and brain waves, 2017 International Conference on Mathematics and Information Technology (ICMIT), Adrar, 2017, 38-42. https://doi.org/ 10.1109/MATHIT.2017.8259693.

[10] Vanus, J.; Smolon, M.; Koziorek, J., et al.: Voice Control of Technical Functions in Smart Home with KNX Technology, Lecture Notes in Electrical Engineering, 330, 2015, 455-462. https://doi.org/10.1007/978-3-662-45402-2_68.

[11] Boder, D. P.: A new apparatus for voice control of electric timers, Journal of Experimental Psychology, 26(2), 1940, 241-247. https://doi.org/10.1037/h0061738.

[12] Rubio-Drosdov, E.; Diaz-Sanchez, D.; Almenarez, F., et al.: Seamless human-device interaction in the internet of things, IEEE Transactions on Consumer Electronics, 63(4), 2018, 490-498. https://doi.org/10.1109/TCE.2017.015076.

[13] Du, G.; Chen, M.; Liu, C., et al.: Online Robot Teaching with Natural Human – Robot Interaction, IEEE Transactions on Industrial Electronics, 65(12), 2018, 9571-9581. https://doi.org/10.1109/TIE.2018.2823667.

[14] Verde, L.; Pietro, G. D.; Sannino, G.: A methodology for voice classification based on the personalized fundamental frequency estimation, Biomedical Signal Processing and Control, 42, 2018, 134-144. https://doi.org/10.1088/1742-6596/979/1/012027.

[15] Rusko, M.; Marian, T.; Darjaa, S., et al.: Influence of noise on the speaker verification in the air traffic control voice communication, Journal of the Acoustical Society of America, 141(5), 2017, 3469-3469. https://doi.org/10.1121/1.4987211.

[16] Gautam, A.; Naples, J. G.; Eliades, S. J.: Control of speech and voice in cochlear implant patients, The Laryngoscope, 129(9), 2019. https://doi.org/10.1002/lary.27787.

[17] Gibbs, W. W.: Build your own Amazon Echo - Turn a PI into a voice-controlled gadget, IEEE Spectrum, 54(5), 2017, 20-21. https://doi.org/10.1080/00016489.2017.1293293.

[18] Ohta, Y.; Kawano, A.; Kawaguchi, S., et al.: Speech recognition in bilaterally cochlear implanted adults in Tokyo, Japan, Acta Oto-Laryngologica, 137(8), 2017, 1-5. https://doi.org/10.1080/00016489.2017.1293293.

[19] Tsai, W. Y.; Barch, D.; Narayanan, V., et al.: Always-on Speech Recognition using TrueNorth, a Reconfigurable, Neurosynaptic Processor, IEEE Transactions on Computers, 66(6), 2017, 996-1007. https://doi.org/10.1109/TC.2016.2630683.

[20] Yazdani, R.; Segura, A.; Arnau, J. M., et al.: Low-Power Automatic Speech Recognition Through a Mobile GPU and a Viterbi Accelerator, IEEE Micro, 37(1), 2017, 22-29. https://doi.org/10.1109/MM.2017.15.

[21] Ihler, F.; Blum, J.; Steinmetz, G., et al.: Development of a home-based auditory training to improve speech recognition on the telephone for patients with cochlear implants: A randomised trial, Clinical Otolaryngology, 42(6), 2017, 1303. https://doi.org/10.1111/coa.12871.

[22] Pak, C. L.; Katz, W. F.: Recognition of emotional prosody in Mandarin: Evidence from a synthetic speech paradigm, Journal of the Acoustical Society of America, 141(5), 2017, 3701-3701. https://doi.org/10.1121/1.4988076.