Study and Implementation of Feature Extraction and Comparison In Voice Recognition

Jiehua Dai Zhengzhe Wei
Linköpings universitetet, Sweden
Email: {jieda119, zhewe844}@student.liu.se
Supervisor: Christoph Schuba

Abstract

With the increasing availability and use of digital cellular and VoIP technology, there has been increased interest in the effects of voice feature extraction and comparison algorithms on speaker recognition systems. After several comparisons among possible speech processing approaches, we designed and developed a speaker recognition system, which is implemented on the MATLAB software package on a Windows operating system environment.

We utilize freely available software to record voice samples of different persons. The samples are analyzed and their representation is used as a user template in the enrollment phase for each user into a biometric authentication system. Furthermore, we implemented the functionality to extract voice features to be able to compare new samples with user templates.

We evaluated our implementation with respect to the FAR (False Acceptance Rate) and FRR (False Rejection Rate) of the system in order to reduce FAR. We found that voice processing subsystem was relatively simple, which profoundly influenced our system performance. Examining the effect of Euclidean Distance is our next step.

1. Background

1.1 Voice

Voice is easy to capture and voice print is an acceptable biometric in almost all societies [1]. Voice may be the only feasible biometric in applications requiring person recognition over a telephone. Voice is not expected to be sufficiently distinctive to permit identification of an individual from a large database.

Moreover, a voice signal is typically degraded in quality by the microphone, communication channel, and digitizer characteristics. Voice is also affected by a person's health (e.g., cold), stress, emotions, and so on. Besides, some people seem to be extraordinarily skilled in mimicking others [1].

1.2 Biometric Systems

Biometric systems are systems that use different methods to recognize humans based upon their physical or behavioral characteristics. It should meet the requirements of universality, distinctiveness, permanence and collective ability.

After obtaining one or more of a person's physical and behavioral characteristics, the IT biometric system get him/her registered. This information is then processed by a numerical algorithm, and entered a database. The algorithm creates a digital representation of the obtained biometric. If the user is new to the system, he or she enrolls, which means that the digital template of the biometric is entered into the database. Each subsequent attempt to use the system, or authenticate, requires the biometric of the user to be captured again, and processed into a digital template. That template is then compared with existing user template in the database to make the matching result. The process of converting the acquired biometric into a digital template for comparison is completed each time the user authenticates to the system [2].

Current technologies have widely varying Equal Error Rates, varying from as low as 60% and as high as 99.9% [1].

1.3 Voice Biometrics

Voice biometrics, meaning speaker recognition, identification and verification technologies should never be confused with speech recognition technologies.

Speech recognition technologies are capable of recognizing what a person is saying without recognizing who the person is. Applications of speech recognition for security purposes or secure transactions are therefore limited [3].

In contrast, speaker recognition, verification and identification technologies can be used to ascertain whether the speaker is the person he or she claims to be.

According to leading voice-based biometrics analyst J. Markowitz, Consultants [4]:

Speaker identification is "the process of finding and attaching a speaker identity to the voice of an unknown speaker. Automated speaker identification does this by comparing the voice with stored samples in a database of voice models."

Speaker verification is "the process of determining whether a person is who she/he claims to be. It entails a one-to-one comparison between a newly input voiceprint (by the claimant) and the voiceprint for the claimed identity that is stored in the system."

Our system should involve both speaker identification and speaker verification.

1.4 Required Software before We Start

A complete voice recognition system includes data preparing tools from outside sources. First we need the software record and save voice samples. Here we choose Microsoft® Sound Recorder V5.1. Wave (.wav) file is voice format we use to save voice samples.

The core development environment is MATLAB. MATLAB® is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows us to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or FORTRAN [5].

In this case, we can use MATLAB's powerful digital signal analyzing and processing to decrease the project computation complexity and reduce our research workload. At the same time, with MATLAB we pay most of our attentions on how to improve the algorithm and system performance, instead of fundamental data processing.

1.5 Basic Constraints

The voice samples should be saved as one kind of criterion data that can adapt to the following process. Here we define basic constraints on voice samples used in our system.

The data type of the voice is wave file. Our system only accepts wave file format.

The voice record should be taken in a quiet environment. The less background noise we have, the better data input our system has.

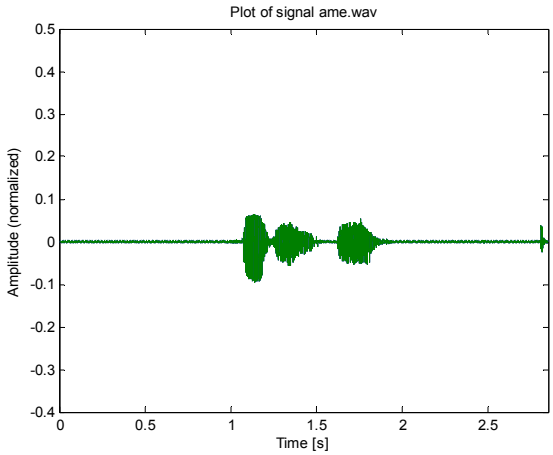


Figure 1. "Who am I?" -- A normal voice sample we expect,

In this project, we try to select an existing word group .Every user should only speak this word group in the process of enrollment and recognition. This makes our system easy to complete the task. Additionally, we can select different word groups, compare their performance and study the word group factor in the voice recognition system.

We have other problems if we use voices as the input of the system. First is the idle part in the beginning of voice samples. Second, if lengths of the same user voice samples are different, we will get different results because of the processing of MATLAB. To get a better result, the word group user read should be approximately 2 seconds in length.

2. User Template

2.1 Data Structure Definition

The structure should be defined on the feature extraction algorithms, so that it can easily compare the user template and new voice samples.

In our design, the user template is a big matrix storing important voice feature values.

2.2 Enrollment

In the enrollment phase, the voice sample of an individual is first recorded by a voice recorder to produce a raw representation of the characteristic.

2.3 Update/Delete, etc.

Voice can vary with age, illness and emotions, so when someone's voice has changed, the original template should be deleted and new samples should be updated as the user template.

3. Extract Features

3.1 Two Factors of Voice

The voice samples can be separated into frames in the process of extract. The size and rate should be set to an appropriate value.

For instance, we can use the following formula to calculate the voice frame size:

m = 100;
n = 256;
l = length(s);
nbFrame = floor((l - n) / m) + 1;
m is the rate , nbFrame is the voice frame size

3.2 Algorithms

Algorithms are used to get features of the voice samples, and then the result can be saved into the data file. Here we use MFCC algorithm.

Mel Frequency Cepstral Coefficients (MFCCs) are coefficients that represent audio. They are derived from a type of cepstral representation of the audio clip (a "spectrum-of-a-spectrum"). The difference between the cepstrum and the Mel-frequency cepstrum is that in the MFC, the frequency bands are positioned logarithmically (on the mel scale) which approximates the human auditory system's response more closely than the linearly-spaced frequency bands obtained directly from the FFT (Fast Fourier Transform) or DCT (Discrete Cosine Transform). This can allow for better data processing, for example, in audio compression. However, unlike the sonogram, MFCCs lack an outer ear model and, hence, cannot represent perceived loudness accurately [6].

MFCCs are commonly derived as follows:

1. Take the Fourier transform of (a windowed excerpt of) a signal
2. Map the log amplitudes of the spectrum obtained above onto the Mel scale, using triangular overlapping windows.
3. Take the Discrete Cosine Transform of the list of Mel log-amplitudes, as if it were a signal.
4. The MFCCs are the amplitudes of the resulting spectrum.

In our system, MFCC.m module includes the algorithm above. Generally, we can take this module like:

Inputs: variables contain the signal to analyze, and the sampling rate of the signal.

Output: variables contain the transformed signal.

3.3 What do we expect

We expect the extract features can uniquely represent the person it should be, and two different persons do not have much in similar in the features we extracted.

4. Comparison Algorithm

The comparison process involves the use of a Euclidean distance. In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, which can be proven by repeated application of the Pythagorean Theorem. By using this formula as distance, Euclidean space becomes a metric space (even a Hilbert space). This is a measurement of how similar two user templates are.

Thus, the Euclidean distance measures the percentage of dissimilar bits out of the number of comparisons made. Ideally, when a user logs in, nearly his entire features match; then when someone else tries to log in, who does not fully match, and the system will not allow the new person to log in.

In our system, the Euclidean distance D between two vectors X and Y is: D = sum((x-y).^2).^0.5

4.1 Input/Output

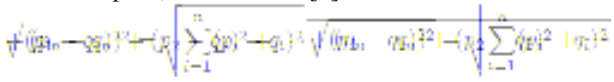
Generally, we can take this module like:

Input: User templates and voice features (x, y): Two matrices whose each column is a vector data.

Output: Comparison results, Euclidean distance (d): Element d(i,j) will be the Euclidean distance between two column vectors X(:,i) and Y(:,j)

4.2 The algorithm

The Euclidean distance between two points $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_n)$, in Euclidean n-space, is defined as [7]:



Two-dimensional distance

For two 2D points, $P = (p_x, p_y)$ and $Q = (q_x, q_y)$, the distance is computed as [8]:

$$\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

Alternatively, expressed in circular coordinates (also known as polar coordinates), using $P = (r_1, \theta_1)$ and $Q = (r_2, \theta_2)$, the distance can be computed as:

$$\sqrt{r_1^2 + r_2^2 - r_1 r_2 cos(\theta_1 - \theta_2)}$$

2D approximations for computer applications

A fast approximation of 2D distance based on an octagonal boundary can be computed as follows. Let dx = | px − qx | (absolute value) and dy = | py − qy | . If dy > dx, approximated distance is $0.41\,dx + 0.941246\,dy$. (If dy < dx, swap these values.) The difference from the exact distance is

between -6% and +3%; more than 85% of all possible differences are between −3% to +3%.

5. Implementation of software

To make a whole system, we need development environments, interfaces, etc. Then we combine these components (feature extrication, user template operations, comparison algorithm and interfaces) together to implement the system.

Our system development environments is: Microsoft Windows XP + VB.net 2005 + Matlab 7.0

The main function is like:

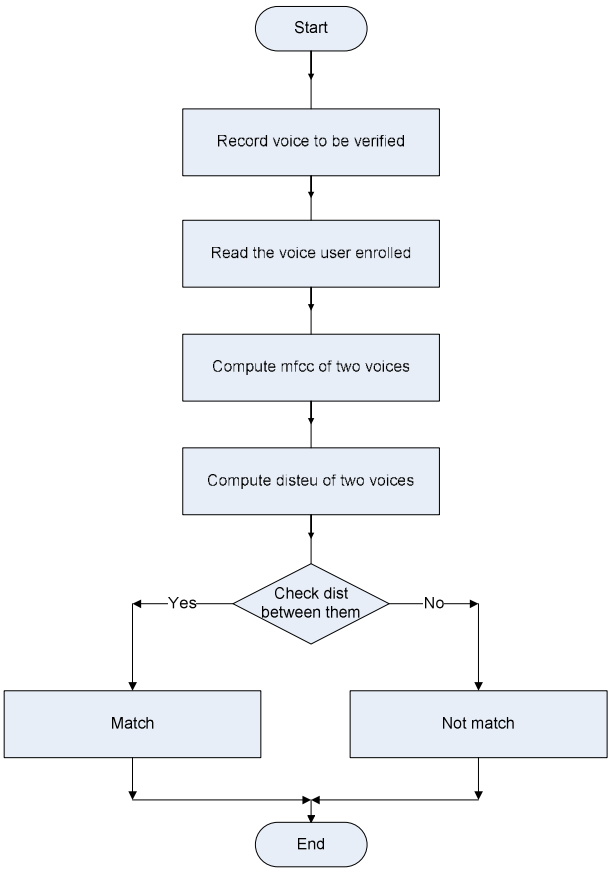


Figure 2. System identify logic flow

6. Results

In this section, we present experiments and results using our voice recognition system. The experiments examine three areas: feature extract distribution, performance using fixed word groups, and combination of different men and word groups.

Here we choose the fixed word group as “You are right.”

6.1 Feature Extract Distribution

In this section, we focus on the feature extract distribution generated by our system.

Let the same person say the same fixed word group twice, and then we have two signals’ maps after we input them into the feature extract module. Figure 3 shows the entire value of these two user templates if we take it from that matrix. Figure 4 shows the key feature we select from the matrix. They will match if most of key features are close enough.

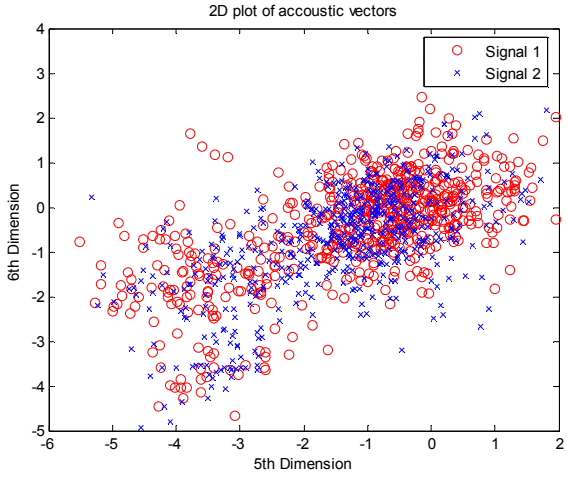


Figure 3: Two user templates values in two dimensions

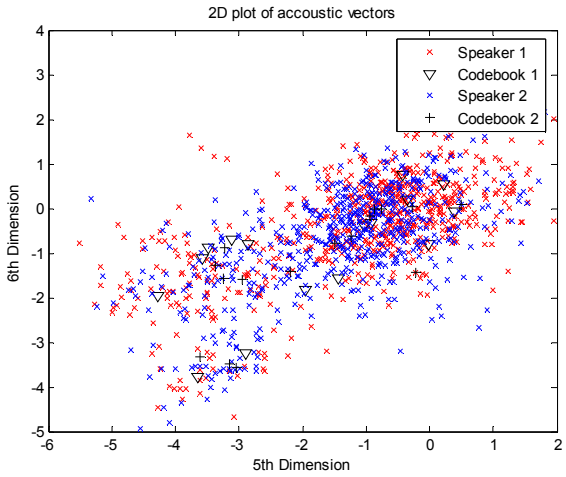


Figure 4: Two user feature distributions in two dimensions

6.2 Evaluation and Comparison

Here we evaluate our system, and compare different speaking word groups.

Table 1: 11 empirical tests of Euclidean Distance with the same content. In each test, we put person A's train voice as the enrolled sample. Let A do the identification, speaking the same content, and then let another person B do it again.

Test 1.	same content same man	same content wrong man
Test 1-1	4.8081	4.8642
Test 1-2	3.7255	5.0761
Test 1-3	4.0012	4.8210
Test 1-4	3.8314	4.9012
Test 1-5	4.1302	4.4503
Test 1-6	4.1073	4.6747
Test 1-7	3.3666	4.2933
Test 1-8	4.0214	4.9735
Test 1-9	4.1976	4.6054
Test 1-10	3.9487	4.7126
Test 1-11	4.1796	4.0374

Table 2: 10 empirical tests of Euclidean Distance with the wrong content. In each test, we put person A's train voice as the enrolled sample, then let A do the identification but with wrong content "I'm right", and let another person B do it again.

Test 2.	wrong content same man	wrong content wrong man
Test 2-1	4.1741	5.5051
Test 2-2	4.1831	6.6019
Test 2-3	5.2193	4.9494
Test 2-4	6.0780	4.6477
Test 2-5	4.4994	5.0492
Test 2-6	4.2247	4.9510
Test 2-7	4.4993	5.1385
Test 2-8	4.1469	4.5923
Test 2-9	4.7603	4.4547
Test 2-10	4.3379	4.8207

After lots of similar tests, we have the FAR is around 10%, and FRR is around 18%.

7. Conclusions

In this paper, we introduced principles of voice recognition, and benefits of using MATLAB in our project. Based on several constraints, we defined basic user template. After compared some existing feature extraction and comparison algorithms, we designed our own algorithms, and implemented them in MATLAB. Finally, we combined these components together to implement the whole system.

It was also shown, though experiments with the contrast approach, that our system could identify the speaker most of the time. We evaluated our system's FAR and FRR and found that its FAR and FRR are worse than other existing speaker recognition system. We believe it's because user voice processing subsystem is relatively simple and needs further improvement.

Future work will focus on better selection of word groups and using speaker-dependent word groups. Additionally we will examine the effect of Euclidean Distance and the use of other acoustic units, such as phones or automatically derived units.

References

[1] J. Campbell, "Speaker Recognition: A Tutorial", Proceedings of the IEEE, vol. 85, No. 9, September 1997.

[2] A. Jain, A. Ross, S. Pankanti, "Biometrics: A Tool for Information Security", IEEE Transactions On Information Forensics And Security 1s, June 2006.

[3] Z. Barzilay, "Voice Biometrics as a Natural and Cost-Effective Method of Authentication", CellMax Systems Ltd. 2007.

[4] J. Markowitz, "Using Speech Recognition: A Guide for Application Developers", Prentice Hall, 1st edition, December 11, 1995.

[5] "MATLAB Help - Introduction", MathWorks Inc. 2004.

[6] M. Slaney, "Auditory Toolbox", Interval Research Corp. 1998.

[7] P. Black, "Euclidean distance", U.S. National Institute of Standards and Technology, 17 December 2004

[8] H. Breu, J. Gil, D. Kirkpatrick, M. Werman, "Linear-time euclidean distance transform algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(5):529–533, May 1995.

[9] D, Maltoni, et.al., "Handbook of Fingerprint Recognition", Springer Verlag, 2003