A VOICE VERIFIER FOR FACE/VOICE BASED PERSON VERIFICATION SYSTEM

R.-Y. Qiao, Y. Choi and J. I. Agbinya
CSIRO Telecommunications and Industrial Physics
Cnr Vimiera & Pembroke Rds, Marsfield, NSW 2122

Abstract

A person verification system based on voice and facial images has been developed within CSIRO Telecommunications and Industrial Physics, Australia, for use in low-to-medium security systems. It provides a unique ID, which is non-intrusive, fast, and has no need for memorising passwords. A stand-alone version of the voice verifier has an error rate of less than 8%, while the face verifier has an error rate of less than 5%. By combining the two modules, an error rate of less than 1% is achieved. This paper describes in detail the method and some of the important practical issues in the implementation of the voice verifier. It also addresses the issue of decision making if the two sub-systems produce contradictory results.

1 Introduction

Computer system security has been a major issue ever since the invention of the computers. Traditionally, people have been relied on passwords to prevent unauthorised entry into their systems and so far the method has worked relatively well. However, with more and more systems requiring passwords, it has become quite inconvenient and difficult to remember many passwords. Some biometrics based person verification systems, such as those based on finger prints and retinal scans, have been developed to overcome this problem and provide higher security. Because they are intrusive and have the connotations of criminal attached to them, people have been reluctant to use them as daily security measures. In contrast, personal face image or voice based systems are non-intrusive, easy to use and widely acceptable. They provide a natural form of accessing not only computer systems but also many other electronic systems as well.

CSIRO Telecommunications and Industrial Physics, Australia, has developed a fast and reliable face recognition system[1]. The system has an equal false-acceptance/false-rejection error rate of less than 5%. As one of our efforts to enhance the system's verification performance, we chose a dual-modality approach by adding a voice-based verifier to the system. This has made the overall system much more robust to different lighting/sound conditions or facial/vocal changes. Under typical office environment where lighting is moderate and ambient noise level is relatively high, an error rate of less than 1% can be achieved by the combined system.

In this paper, we focus on the voice verifier and the problem of how to make decisions when the two sub-systems are combined. For interested readers, reference [1] provides detailed information on our face verification system.

2 System Description

The voice verifier uses randomly prompted digits as voice password and a Hidden Markov Model (HMM) as the classifier. Compared with a user-chosen password, there is no need to remember the password because each time a user tries to log into the system, s/he will be asked to speak a different randomly chosen digit sequence. Randomising the digit sequence can also increase the system security because it makes entries with recorded voice very difficult. The system consists of 5 modules, and the diagram is shown in Figure 1. The output of this system is a confidence measure about the user's identity, and is combined with the output from our face verifier to make the final decision.

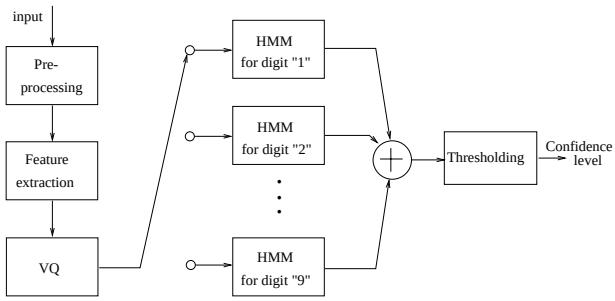

Figure 1. System diagram of the voice verifier

Not shown in the diagram is a silence removal module before the pre-processing. This module has been found to be critical to the system performance in noisy environments. Noise and non-speech sounds will be removed and each digit utterance segmented into a fixed length sequence, which guarantees the same number of symbols (indices into the vector quantisation codebook) for all the digits. The reason for using a fixed-length block for each digit is that we can linearly combine the classification scores without the need for time alignment between digits.

2.1 Pre-processing

Input speech is sampled at 11.025kHz with 16-bit resolution, the rate chosen for programming convenience on a MS-Windows based PC. In fact, any other sampling rate between 8 and 16kHz could be used simply because most of energy of human speech is concentrated below 4kHz frequency range. Higher sampling rates than 16kHz would only increase the system computational load with only minor improvement of system performance. A band-pass filter is used to limit the voice bandwidth to between 50 and 3400kHz. This band-pass filter also serves the purpose of removing non-speech sounds outside that frequency range.

The input speech is also pre-emphasised with a high-pass filter which has a transfer function of $1 - 0.95z^{-1}$. This has the effect of spectral flattening and increasing the signal to noise ratio at high frequency range to better preserve the speaker's voice characteristics.

2.2 Feature Extraction

Speech is generally piecewise-stationary within a duration of around 20 ms. For each block of 256 samples (23 ms), the input speech samples are windowed with overlapping Hamming windows, and some features are extracted. The commonly used ones are the linear prediction coefficients (LPCs), LPC-derived cepstral coefficients and Mel-frequency cepstral coefficients (MFCC). Other speaker-specific features such as pitch, speech intensity, formant frequencies can also be used for verification. But due to the difficulties of measuring these parameters, they are rarely used. Therefore, by far the most prevalent features have been the short-term spectrum-based ones. Research has shown that among all the features for speech recognition, the MFCC gives the best performance[2].

The MFCC is calculated as following:

$$\theta_{tp} = \sum_{q=1}^{Q} \log S_{tq} \cos\left[p(q - \frac{1}{2})\frac{\pi}{Q}\right]$$

where Q is the number of triangular filter banks, S_{tq} the energy in each logrithmic band, p the MFCC order and t the block index. A total of 12 MFCCs are calculated for each block of 256 samples of speech.

2.3 Vector Quantisation

Features extracted in such a way are normally continuous-valued, and are represented as feature vectors. It is necessary to transform these vectors into a set of orthogonal base vectors, in order to reduce the total number of vectors the classifier has to work with. Vector quantisation (VQ) serves such a purpose. Without VQ, all parameters within the classifier have to be approximated by using a mixture of continuous probability density functions (PDF) instead of numbers, and parameter estimation in such a situation becomes much more complex.

A codebook with 128 code vectors was designed using the LBG algorithm[3]. Each vector is a block of 12 speech samples, corresponding to the 12 MFCCs obtained in the feature extraction stage for each frame of data. Output from the quantiser is the index to the codebook. This index is used as the observation symbols (1-128) of the HMM classifier.

2.4 Hidden Markov Model based Classifier

There are a few methods that can be used as a classifier, but most of them can be classified into two categories, namely the dynamic time warping (DTW) based method and the hidden Markov model (HMM) based method. The DTW is basically a template matching method, in which some distances between two vectors with different time scale are measured. Systems using HMM have shown greater flexibility and generally perform better than DTW-based ones. But for systems where computational power is limited, the DTW approach may be preferred due to its simplicity.

In contrast to this template-matching method, the HMM approach uses statistics derived from speech feature vectors to score the likelihood of a given input feature vector. Each person registered in the system has his/her own models trained during the initial training sessions. Decision of either an acceptance or a rejection of the user will be made by comparing this likelihood with a pre-set threshold stored in the database with the models. In our system, we use only the 9 single digits (1-9) to form a password. Each of the 9 digits is modelled as a separate HMM, as shown in Figure 1. Because we know which digit is prompted (or spoken), we can test the spoken digit using the model of that digit, without having to go through all the models to find out which model produces the maximal score.

For each digit, a 6-state left-right type HMM is built for each person. Our test results have shown that even though the system performance is not very sensitive to the number of states used in the HMMs, a 6-state HMM performs slightly better than others[4]. A single state could be either a phoneme in a digit, or an observation interval of 10-15ms (Bakis model). For a discrete density HMM, 512-1024 codewords or symbols are required to give satisfactory performance.

For the training of the HMMs, batch training with multiple sequences is used. We have found that 10 or more repetitions of each digit is sufficient for a relatively good performance with the 6-state HMMs. We can reduce the number of digits required for training by reducing the number of states used with minimal reduction in the performance.

When starting to build a model from scratch, initial values to all the parameters have to be assigned. Since the re-estimation procedure (the Expectation-Maximisation method) can only guarantee at its best a locally optimal solution (local minimum of the PDFs), initial estimates of the HMM parameters are crucial for a good model. Rabiner[4] has suggested a few ways of getting the initial estimates. We have found, however, that using a simple initialisation procedure where all parameters are set to random values works just as well for our purpose. All parameters, of course, have to be guaranteed non-negative.

2.5 Start/End Point Detection

To correctly pick up a spoken digit, a good start/end point detection method is needed. Ideally, it should remove all the silence and non-speech sounds from the input. Silence simply adds unnecessary computational load to the system and could also affect the classification performance in the training stage, while non-speech sounds bear no speaker's voice information and will severely increase the verification error rate.

One of the difficulties of the Barkis model is time alignment of two utterances at different speaking speeds from one speaker. Since each symbol (codebook index) corresponds to a speech frame (in this case, 256 samples with overlapping of 128 samples), variations in speaking speed will affect the number of frames, hence the number of symbols within each word (digit). The score produced by the classifier is a measure of the likelihood that the model generates the symbol sequence. Hence different sequence length will inevitably result in different scores, even though the two words are the same, spoken by the same speaker. Clearly there is a need to combat this problem so that the scores in such a situation are close to each other.

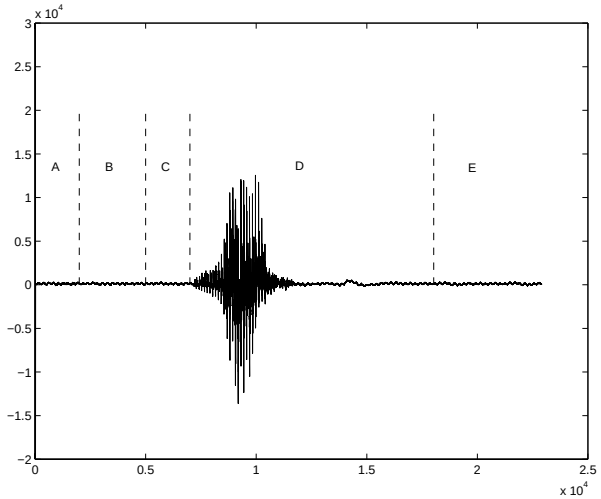

Figure 2. Start point detection

Figure 2 shows how a digit utterance is separated from the silence and non-speech sounds. Non-speech sounds normally have a very different frequency spectrum from speech, and can be mostly removed by band-limiting the input signal to voice band (100-3400Hz) in the pre-processing stage. The pre-processing can also

filter out the possible DC bias caused by the recording devices. In our voice verifier, we record each digit with a fixed length of at least 2 seconds and prompt digits in such a way that there are always a minimum of 5500 samples of silence before the utterances. After band-limiting and pre-processing the recorded speech, we discard the first block of samples (block A in Figure 2) to remove the possible spikes caused by the recording hardware. The next block, block B, calculates the noise energy level, which will be used as the threshold for determining the start point of the digit. We continuously monitor the magnitude of the recording samples. If the magnitude becomes greater than Δ times of the noise level, we mark the sample as the start point of that digit and take a fixed length block of samples as the training or testing data and store it in a file.

Utterances picked in this way may still contain non-speech sounds and silence, especially at the beginning of the digits. We have found that the duration of a digit rarely goes over 4500 samples. So this value is used for segmenting the digits during the training/verifying stage. In a further effort to accurately pick up a digit, we search through the stored digit and find the peak position of the digit. Then we go backward from this point and measure the total energy in 100-sample blocks. If the energy drops below certain value, then the middle point of that block will be marked as the starting point of that digit.

This two-stage approach has been proved to work quite well, making the system very robust to ambient noises.

3 Program description

The program has been developed as a 32-bit dynamic link library (DLL) for Microsoft Windows 95/NT, so it can be easily incorporated into our person verification system[1]. The online DLL interface is shown in Figure 3. To use the system, a user has to type in his/her user name, and then speak the displayed digits to the microphone. For the combined system with face verifier, the face images are taken simultaneously while the user is speaking the password.

To train the models, the user follows the same procedure as with verification. The user can choose to re-train his/her models if they already exist or to discard them and create new models.

Figure 3. The online voice verifier interface

4 Decision Making

Each digit spoken by a user produces a classification score for that digit. When verified against a user's own models, the utterance should result in a lower score (or higher probability) than with other users' models. Shown in Figure 4 are the scores when a user (User 1) speaks the digits "1", "2" and "3" and verifies against other users. All scores are normalised by the user's own test score. There are quite large variations in the values between different digits as well as users. Identifying user 1 from the rest can be easily done by finding the lowest total scores. The separations between scores can be made greater with different combining method. Figure 5 shows the difference between combining all the scores linearly and multiplying them together to get the total score. Clearly the multiplication method gives better separation of users.

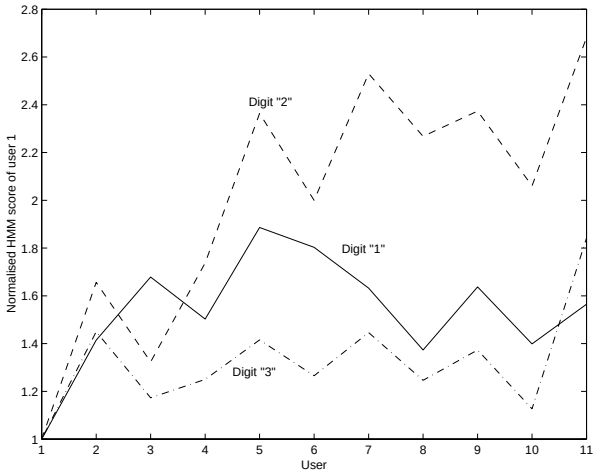

Figure 4. One user's utterance is tested against other users' models

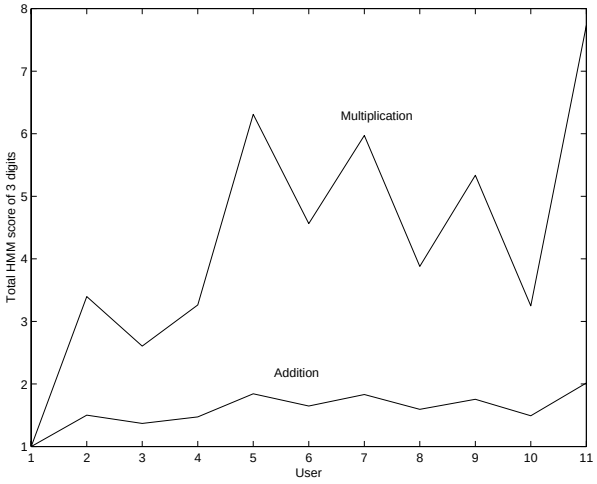

Figure 5. Different methods used for combining the scores of different digits

In a real situation where a decision has to be made online, setting a threshold to separate the true user from impostors is not a trivial issue. Normally we choose the threshold that gives an equal false-acceptance false-rejection error rate. Under different circumstances, however, it may be preferable to use a lower threshold which gives lower false-acceptance rate than false-rejection to provide a higher security, or vice versa. Even though different users produce different verification scores, we have found that dynamic thresholding among users did not provide any better performance than a simple hard thresholding method. This could be due to the passwords being random digits, which result in different scores with each different combination.

Since the voice verifier forms part of the person verification system, final decision as to whether a user is accepted or rejected is based on the overall scores of the whole system. How to combine the scores from the voice and face verifiers will largely affect the overall system performance. We normalise the distance between the test score and the threshold to the range of [-1,1], with the absolute value being a confidence measure toward either acceptance or rejection. Then we make the final decision based on

the total combined score of the two sub-systems. If the combined score is non-zero, the user is accepted or rejected, depending the sign of the score. When the score is zero, however, the voice and face verifiers are equally confident of their decisions, then ambiguity occurs. In this case, we can either ask the user to try again, which is the simpler solution, or introduce a weighting factor λ to one of the sub-systems and weight the other by $1 - \lambda$. This factor allows us to take into account the environment conditions. If the environment condition is more favourable to voice recorder than to the camera, e.g. quiet but poor lighting, we can weight the voice score more heavily than the face score when combining them. Otherwise, we will emphasise the face score when making the final decisions.

5 Results

Shown in Figure 6 are the false-positive (acceptance) false-negative (rejection) percentages for a password length of 5. Figure 7 shows the same performance indicator when the password length is 7 digits. We obtain the false-negative data by testing a user's speech against his/her own models. In total, we tested 10 times for each user (12 users) using a random password (same length, different permutations taken from the 9 digits) each time. For the false-positive scores, a random password from each of the 12 users is tested against all other users' models. Ten repetitions of each digit are used for training the 6-state HMMs. The training and verification are done in a relatively noisy office with other people talking and fan-noise from PCs and air-conditions in the office. An equal error rate of 8% is achieved with a hard threshold of 884. If the password length is increased to 9, the error rate will be reduced to below 5%, as shown in Figure 8.

6 Conclusions

We have developed a voice verifier using HMM as part of a person verification system based on face image and voice. The voice verifier can be used as a stand-alone verification system which would have a success rate of more than 92% in relatively noisy environment, if the password length is 5 or more. The password is a random combination of the 9 digits, and changes each time a user tries to login. This makes the system more secure than those systems using a fixed password. Because of the method used for detecting the start/end points of digits and removal of noises and non-speech sounds, the system is very robust against such problems.

The voice verifier has been integrated with the face verifier developed within this division. A specific procedure was developed to combine the scores from the two sub-systems together and make the final decision. The overall system performance is better than 99% for equal rates of successful acceptance and rejection.

References

[1] G. T. Poulton, N. A. Oakes, D. G. Geers, R.-Y. Qiao, M. D. S. Seneviratne, N. E. Frampton, Y. Choi and J. I. Agbinya, "The CSIRO PC-CHECK System", Submitted to the *7th Intl. Conf. on Audio and Video based Biometric Person Authentication (AVBPA'99)*, Washington, USA, March, 1999.

[2] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Trans. ASSP*, Vol.28, No.4, pp.357-366, August 1980.

[3] Y. Linde, A. Buzo and R. M. Gray, "An Algorithm for Vector Quantiser Design", *IEEE Trans. Commun.*, Vol. COM-28, pp.1551-1588, Jan. 1980.

[4] L. Rabiner and B.-W. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

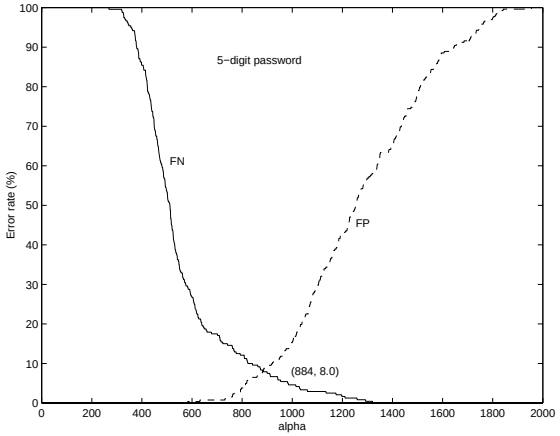

Figure 6. False-positive false-negative scores with 5-digit password

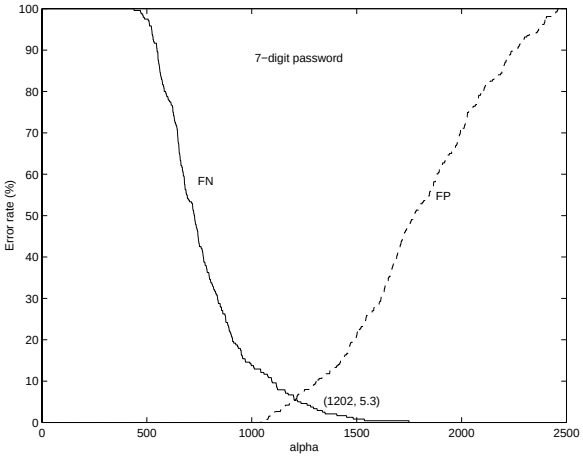

Figure 7. False-positive false-negative scores with 7-digit password

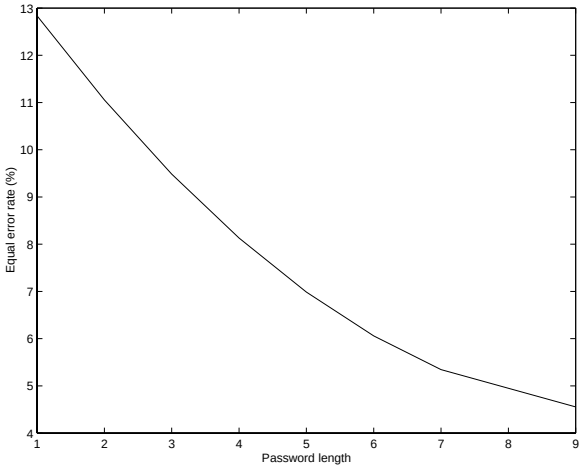

Figure 8. Equal false-positive false-negative error rate with different password lengths