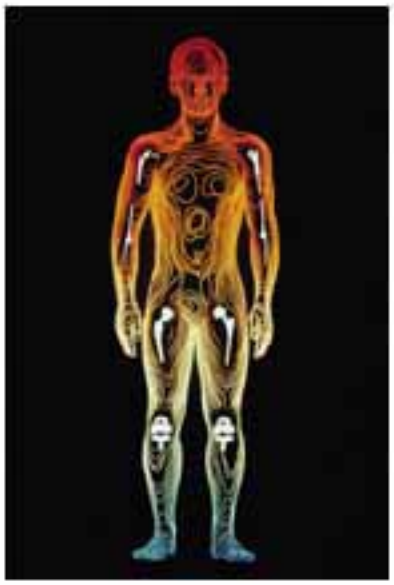
Voice based Biometric Security System



Abhishek Mitra
Saurabh Bisht
Vikas Ranjan

Acknowledgment

We express our gratitude and deep-felt thanks towards our esteemed guide,
Dr. Sudip Sanyal for his able guidance, which enabled us to complete our project.

We are extremely grateful to Prof. ASR Murthy for providing us with valuable suggestions and feedback.

We take this opportunity to thank our colleagues who provided us with valuable suggestions, and also lent their voices for analysis and testing during the course of the project, thus resulting in the successful completion.

Project Details

AIM: Develop a Biometric Security System, which used the human voice as a distinguishing feature between various users.

Project Type:

Mini Project worth three credit hours.

Students Involved:

Abhishek Mitra: 19991003
Saurabh Bisht: 19991043
Vikas Ranjan: 19991054

Duration:

Sixth Semester, January – May 2002.

Software / System Involved:

MATLAB v6.1 and GUI development tool.
Pentium III 866 Mhz, 256 MB RAM running Windows XP.
Stereo Sound Card.
Headphones.
Standard Microphone.

Introduction

Biometrics

Biometrics refers to the automatic identification of a person based on his/her physiological or behavioural characteristics. This method of identification is preferred over traditional methods involving passwords for various reasons

(i) The person to be identified is required to be physically present at the point-of-identification.
(ii) Identification based on biometric techniques obviates the need to remember a password.
(iii) Proxy methods of impersonation will fail.

By replacing passwords, biometric techniques can potentially prevent unauthorized access to or fraudulent use of ATMs, cellular phones, desktop PCs and computer networks. Moreover biometric systems cannot be rendered to forging.
Various types of biometric systems are being used for real-time identification; the most popular are based on face recognition and fingerprint matching. Other biometric systems utilize iris and retinal scan, speech, facial thermograms, and hand geometry. An important issue in designing a practical system is to determine how an individual is identified. Depending on the context, a biometric system can be either a verification (authentication) system or an identification system.

Speech

Speech is the generic name given to sounds, which carry language content. Speech is produced by the excitation of an acoustic tube called the vocal tract, which is terminated on one end by the lips and on the other end by the glottis, manifesting as a longitudinal compression wave. On conversion to digital form a speech signal will be a one-dimensional time varying signal. Speech signals are bandlimited too, most information is found at a finite frequency range.

Speech is generated in three basic ways.
1) Voiced sounds
2) Fricative Sounds
3) Plosive sounds

In the unvoiced fricative sounds, the energy is concentrated high up in the frequency band, and quite disorganized (noise-like) in its appearance. In other unvoiced sounds, e.g. the plosives, much of the speech sound actually consists of silence until strong energy appears at many frequency bands, as an "explosion".

All these sources act as a wide band excitation to the vocal tract, which can be modeled as a slow time varying filter. The vocal tract is characterized by its natural frequencies (formants), which correspond to resonances in the sound characteristics of the vocal tract. Since the vocal tract changes shape slowly during speech, it is reasonable to assume constant characteristics over a time interval of the order of a few milliseconds.

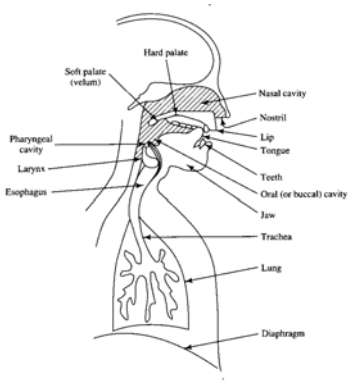


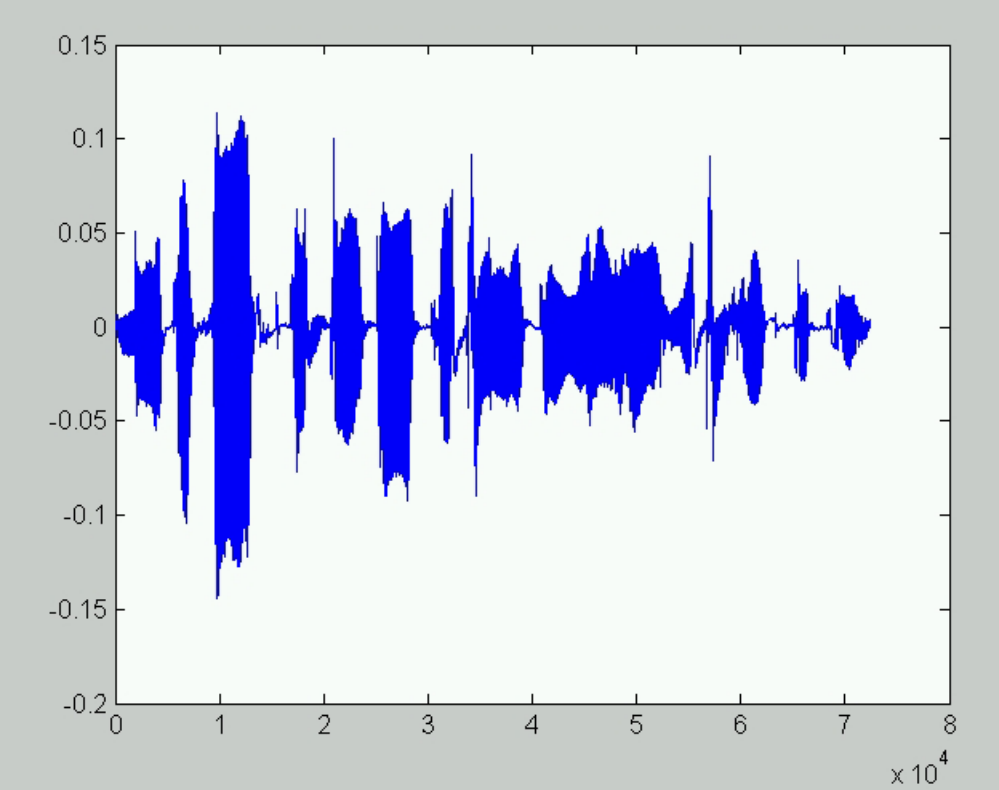
Figure: Human Speech Production System

Speech Analysis

Many techniques are used to analyze a speech waveform, among them a few important ones are enumerated below.

OSCILLOGRAM (WAVEFORM)

Physically the speech signal is a series of pressure changes in the medium between the sound source and the listener. The most common representation of the speech signal is the oscillogram, often called the waveform. In this the time axis is the horizontal axis from left to right and the curve shows how the pressure increases and decreases in the signal. However, a suitable structure is extremely difficult to extract from the mass of information in the intensity waveform. This difficulty motivates us to search for some transformation of the raw intensity waveform into a different representation where the important structure is easier to identify and the enormous amount of variability is reduced.



FUNDAMENTAL FREQUENCY (PITCH)

Another representation of the speech signal is the one produced by a pitch analysis. Speech is normally looked upon as a physical process consisting of two parts: a product of a sound source (the vocal chords) and filtering (by the tongue, lips, teeth etc). The pitch analysis tries to capture the fundamental frequency of the sound source by analyzing the final speech utterance. The fundamental frequency is the dominating frequency of the sound produced by the vocal chords. This analysis is quite difficult to perform. Several algorithms have been developed, but no algorithm has been found which is efficient and correct for all situations. The fundamental frequency is the strongest correlate to how the listener perceives the speaker's accent and stress.

SPECTRUM

According to general theories each periodical waveform may be described as the sum of a number of simple sine waves, each with a particular amplitude, frequency and phase. The spectrum gives a picture of the distribution of frequency and amplitude at a moment in time. The horizontal axis represents frequency, and the vertical axis amplitude. If we want to plot the spectrum as a function of time we need a way of representing a three-dimensional diagram, one such representation is the spectrogram. Various speakers have peaks at certain frequencies, resulting in varied speech qualities.

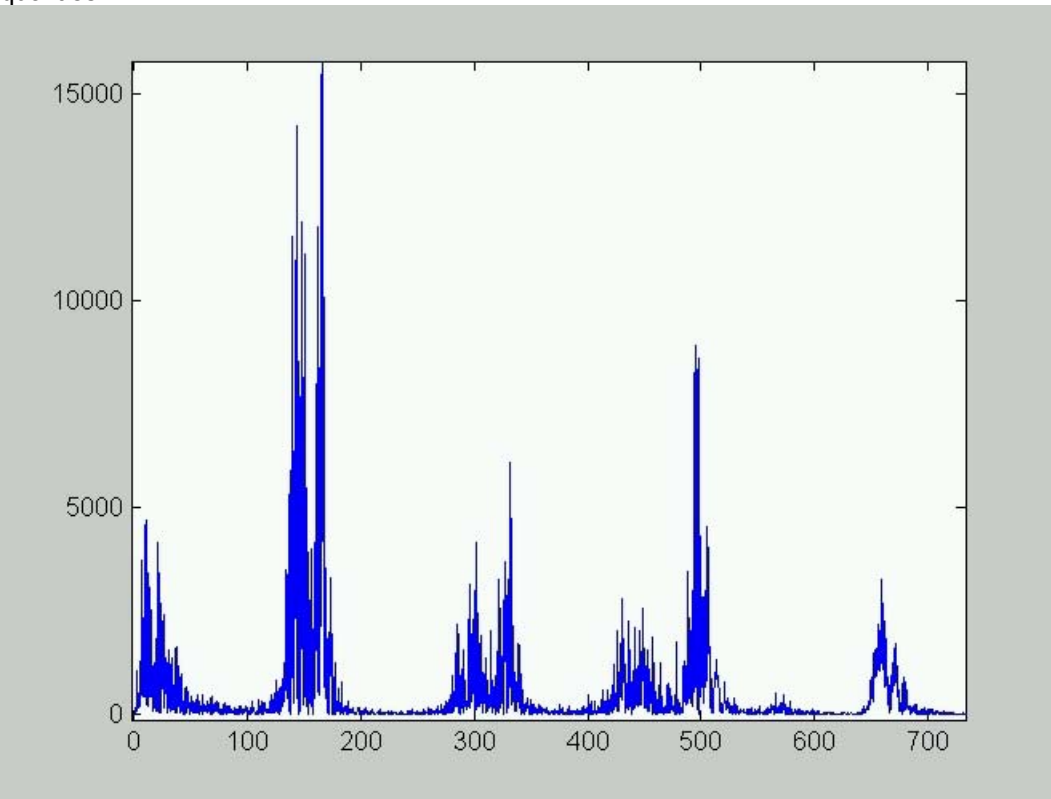


Figure: Spectrum of an user, speaking the sentence ”MICROSOFT, COMPUTER, Triple I T, Yahoo”

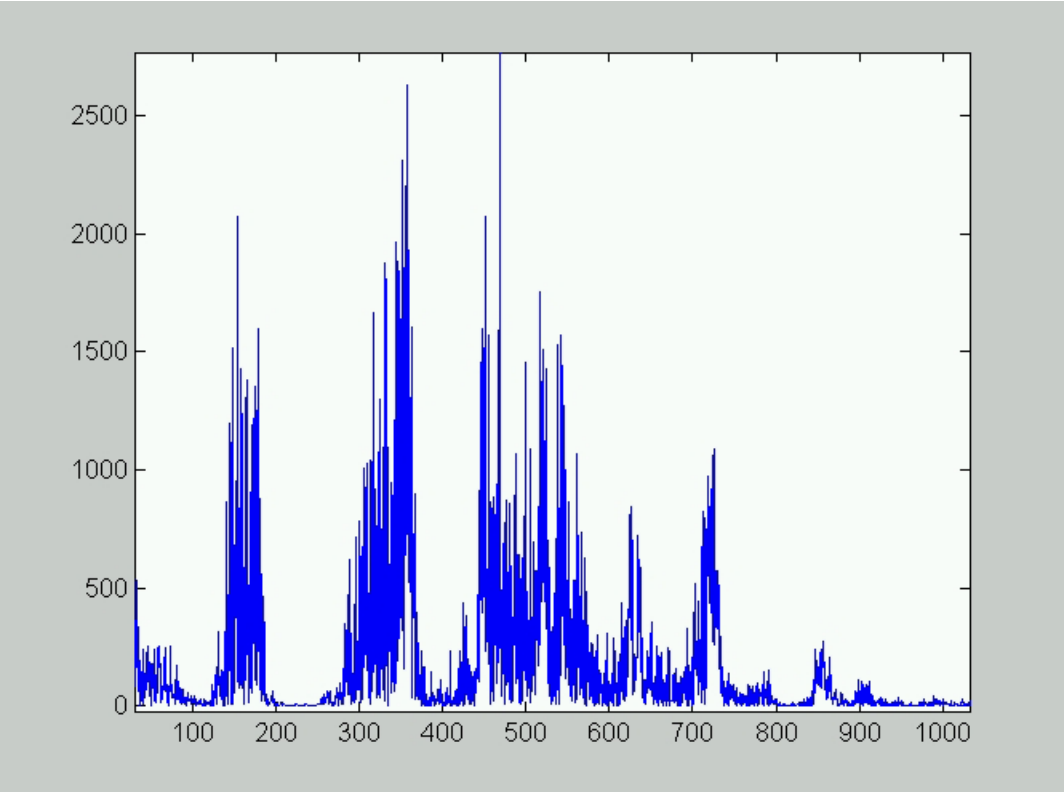


Figure: Spectrum of another user speaking the same sentence, note the varied frequency contents, as compared to the previous spectrum.

SPECTROGRAM

In the spectrogram the time axis is the horizontal axis, and frequency is the vertical axis. The third dimension, amplitude, is represented by shades of darkness. Spectrogram can be considered as a number of spectrums in a row, looked upon "from above", and where the highs in the spectra are represented with dark spots in the spectrogram. From the picture it is obvious how different the speech sounds are from a spectral point of view. The voiced sounds appear more organized. The spectrum highs (dark spots) actually form horizontal bands across the spectrogram. These bands represent frequencies where the shape of the mouth gives resonance to sounds. The bands are called formants. The positions of the formants are different for different sounds.

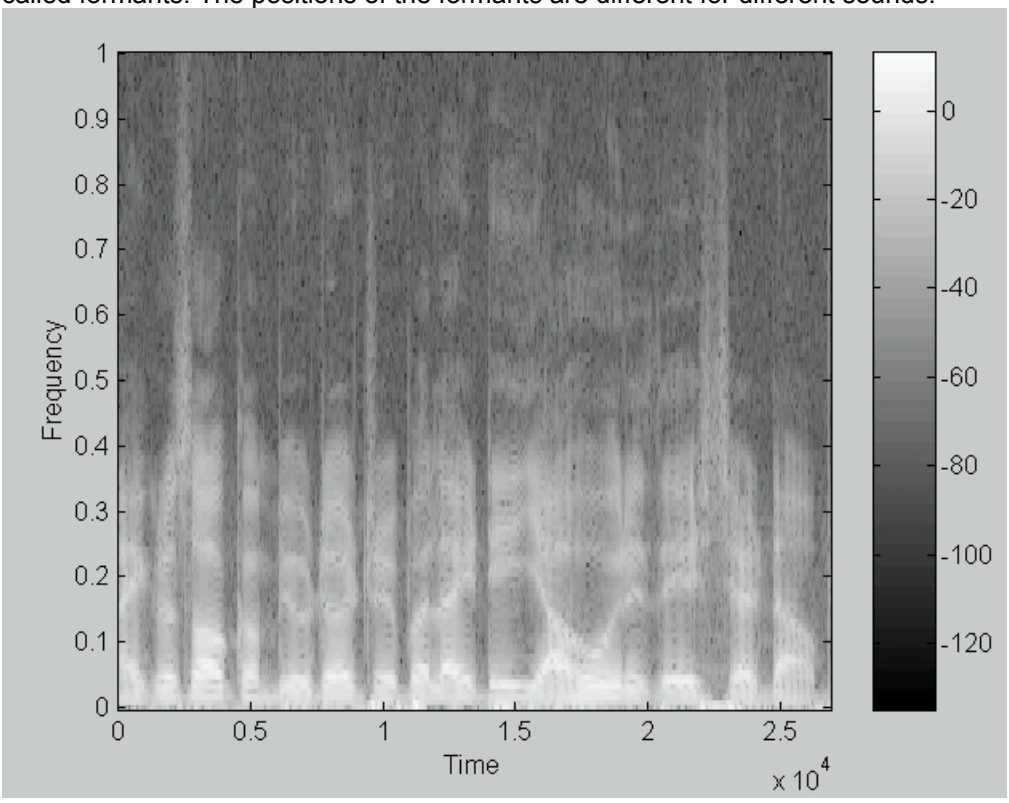


Figure: A spectrogram for an user speaking the sentence "MICROSOFT, COMPUTER, Triple I T, Yahoo". Note the high amplitudes at the lower end of the frequency ranges. The maximum frequency is 11KHz for this spectrogram.

Cepstrum

The cepstrum is a method of speech analysis based on a spectral representation of the signal. One way to think of speech is as a signal being filtered by the mouth cavity. Assuming that the actual speech (S) one tries to produce is the same for all people, the signal that comes out of the mouth and into a data recorder is that signal filtered by the person's voicebox and throat. If we let v represent this filtering, we can write what we record, r, as the convolution of v and s,

```
r= v*s;
```

We need to deconvolve the vocal tract response and the source signal, thus obtaining the fundamental frequency of the speech.

If we move the to the frequency domain we would have:

R = V S

Where R, V, S are the Fourier transforms of r, v, and s respectively. Since we agreed that s is the same for all people, to be able to extract v (or V), there is a need to take the logarithm on both sides to separate the variables.

log R = log V + log S

Thus, an optimal thing for us to compare from sample to sample is this log R quantity instead of just R because the V and S information are combined additively instead of multiplicatively. This type of analysis is known as cepstral analysis. As FFT generates both the real and imaginary parts, we only take the magnitude of each FFT component and calculate the logarithm before taking the inverse FFT. For a feature vector x, the real cepstrum ‘c’ may be calculated by the following formula, [c = real (ifft (log (abs (fft (x)))))].

Features and Phonemes

If we feed an entire phrase into a device to compute the cepstral coefficients, we would get frequency information about the entire sample. This isn't really what we want as we are interested in minute variations in the speech, i.e. we would like to identify features that occur in the short run. Specifically, we would like to identify the phonemes of an individual.
It takes a person about a tenth of a second to utter the word ``hi.'' But ``hi'' is made up of two phonemes, a ``hhh'' sound and an ``iiii'' sound, and we want to extract frequency information from both of them. Clearly, we must chop up the sample into small sections and perform cepstral analysis on these small parts. We call our short-term cepstral vectors as ``feature vectors''.

Security

The speaker-specific characteristics of speech are due to differences in physiological and behavioural aspects of the speech production system in humans. The main physiological aspect of the human speech production system is the vocal tract shape. The vocal tract is generally considered as the speech production organ above the vocal folds.

Verification vs. Identification

Speech recognition, verification or identification systems work by matching patterns generated by the signal processing front-end with patterns previously stored or learnt by the systems.

Voice based security systems come in two flavours, *Speaker Recognition and Speaker Verification*. In Speaker recognition voice samples are obtained and features are extracted from them and stored in a database. These samples are compared with various other stored ones and using methods of pattern recognition the most probable speaker is identified. As the number of speakers and features increase this method becomes more taxing on the computer, as the voice sample needs to be compared with all other samples stored. Another drawback is that when number of users increase it becomes difficult to find unique features for each user, failure to do so may lead to wrong identification.

Speaker Verification is a relatively easy procedure wherein a user supplies the speaker's identity and records his voice. The goal of speaker verification is to confirm the claimed identity of a subject by exploiting individual differences in their speech The features extracted from the voice sample are matched against stored samples corresponding to the given user, therefore verifying the authenticity of the user. In most cases a password protection accompanies the speaker verification process for added security.

It is possible to expand the number of alternative decisions from *accept* and *reject* into *accept*, *reject* and "*unsure*". In this case the system has a possibility to be "unsure". If the system is "unsure", the user could be given a second chance.

DECISION \ USER	*genuine user*	*imposter*
accept	OK	false acceptance
reject	false rejection	OK

Figure: The decision matrix for the system.

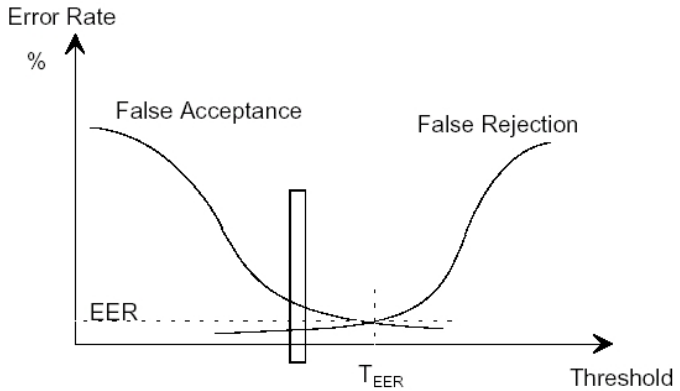


Figure: Threshold selection for minimizing errors in speaker verification. Our system needs to work in a small window, thus rendering the process as a sensitive one.

Text dependent / Text Independent Systems

In text-dependent speaker verification, the decision is made using speech corresponding to known text, and in text-independent speaker verification, the speech is unconstrained.

Various typed of systems in use are:

1) **Fixed password system**, where all users share the same password sentence.
This kind of system is a good way to test speaker discriminability in a text-dependent system
2) **User-specific text-dependent system**, where every user has his own password.
3) **Vocabulary-dependent system**, where a password sequence is composed from a fixed vocabulary to make up new password sequences.
4) **Machine-driven text-independent system**, where the system prompts for an unique text to be spoken.
5) **User-driven text-independent system**, where the user can say any text he wants.

The first three are examples of text dependent systems while the last two are text independent systems. We employed the first system, due to ease of implementation.

Intra-speaker and inter-speaker variability

It is apparent that most voices sound different from each other. It is not so apparent that one single person's voice is likely to sound a bit different from time to time. In the case of a person having a bad cold, it is, however, obvious. The variation in voices between people is termed inter-speaker variability, and the variation of one person's voice from time to time is called intra-speaker variability.

Speaker Recognition System

The problem of a speaker recognition system is to identify a speaker based on his voice given knowledge of a set of speakers through examples. Two of the great difficulties in such a system are identifying features in a known person's voice and searching new samples for these features.

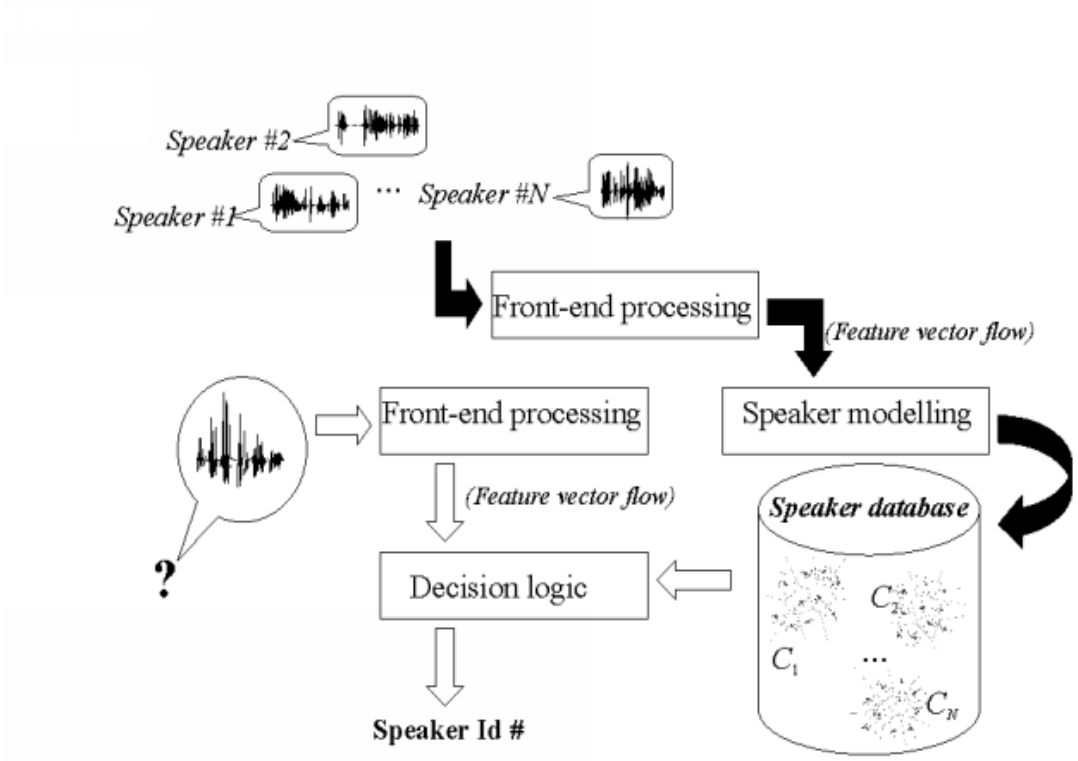


Figure: A typical Speaker Recognition / Verification System.

Speaker recognition is usually a general name referring to two different subtasks: *speaker identification* and *speaker verification*. Referring to the Figure above, we can see that a speaker recognition system is composed of the following modules:

Front-end processing the "signal processing" part, which converts the sampled speech signal into set of *feature vectors*, which characterize the properties of speech that can separate different speakers. Front-end processing is performed both in *training*- and *recognition* phases.

Speaker modelling, this part performs operations on the feature data by modelling the distributions of the feature vectors.

Speaker database, the speaker's feature vectors are stored here.

Decision logic, makes the final decision about the identity of the speaker by comparing unknown feature vectors to all models in the database and selecting the best matching model, or comparison with a selected user, as may be the case.

Front End Processing

Acquisition

Speech signature data acquisition: A database of speech files is maintained for various users. In our system we use four speech files, each containing the spoken equivalent of the sentence **"MICROSOFT, COMPUTER, Triple I T, Yahoo"**. A user should store samples at different times of the day rather than all at one go to get good variability. As people tend to speak in a very prototyped manner if they have to repeat the same thing many times in a short period and this will bias the training data. Moreover the speaker should speak clearly into the microphone without any specific emphasis on any particular word. The recording should be done in a situation when there is very less ambient noise. We can sample the original time signal, which is real and continuous, and use instead a vector of values discretized both in time and amplitude. We store the file in wave file format (PCM), with a sampling rate of 22KHz sampling rate and 16-bit resolution.

Microphone Issues

All microphones measure something roughly proportional to the intensity waveform of the sound or its first derivative by transducing the movement of the air into an electrical voltage using their diaphragms. The type of the microphone used and its distance from mouth also affects the response of the system. It has to be kept in mind that the same microphone has to be used throughout the experiment. For data acquisition, a standard sound card and a single connected microphone (monaural) suffice for the job. However, each individual microphone exhibits a slightly different and usually mildly nonlinear transformation from the true pressure signal to its output voltage. Furthermore, ambient noise from other sources and effects of the medium in which the sound is traveling mean that the pressure wave at the microphone is never the same as that which left the speaker's mouth.

Processing

Once the sample is obtained, we need to remove any dead air at the beginning and at the end. Post silence removal, the sample is normalized so that the waveform amplitude varies between 0 and 1. We break up the voice sample into small frames each 20 – 40ms long with an overlap of 50% i.e. 10 ms. Each frame captures certain part of the speech, which is now windowed using a Hamming Window.

Windowing

A Hamming window lays more emphasis on the center of the frame, and lesser emphasis on the edges, thus minimizing the spectral distortion. The n point hamming windows is generated by the following equation. **w(n) = (54 - 46*cos(2*pi*N/(n-1)))/100**;

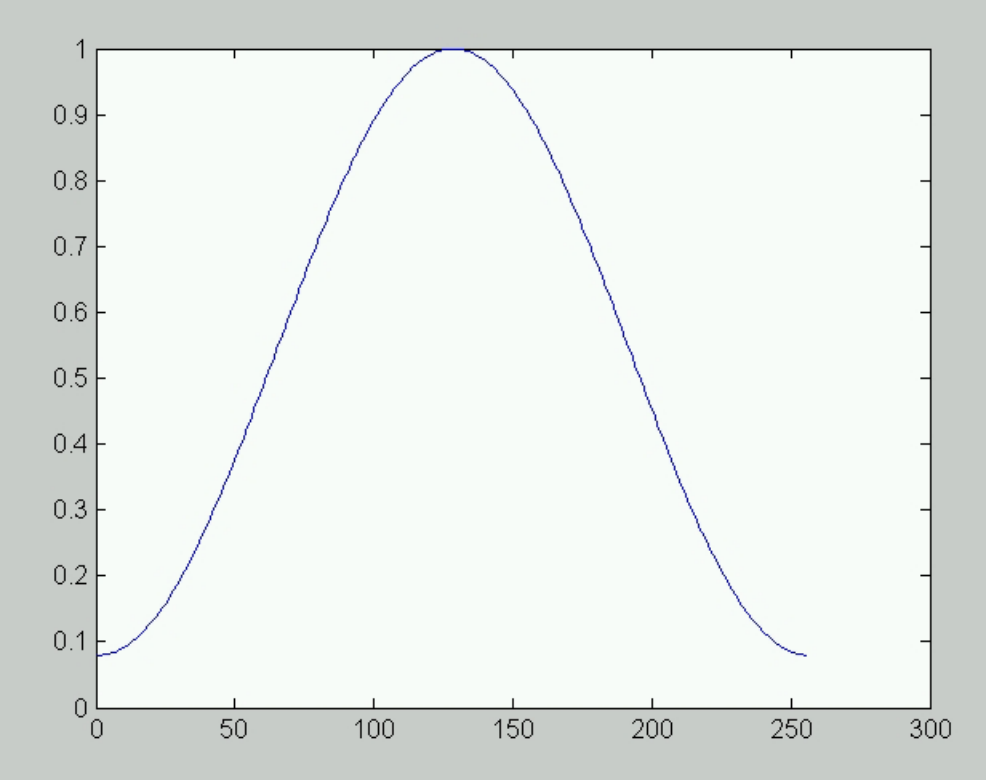


Figure: a 256-point hamming window. At 22KHz a 256 point window is used for an 11.6 msec frame.

Fast Fourier Transform

In this step the fast Fourier of each resultant frame after windowing is performed, thus obtaining the frequency domain representation (spectrum). The spectrum is now filtered using Mel spaced filter banks.

$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)}$$

X(k) = fft(x);
X(k) is the Fast Fourier Transform of frame x(1 to n).

$$\omega_N = e^{(-2\pi i)/N}$$

ω_N is the complex nth root of unity.

Mel Filtering

As mentioned above, psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency, *f*, measured in Hz, a subjective pitch is measured on a scale called the Mel scale. The *mel-frequency* scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz, much similar to the perception model of our ears. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 Mels. Therefore we can use the following approximate formula to compute the Mels for a given frequency *f* in Hz: **mel (f) = 2595 * $\log_{10}$(1+f/700)**

This filter bank is applied in the frequency domain, i.e. on the spectrum calculated in the previous step. A nice way of thinking about this Mel filter bank is to view each filter as a histogram bin (where bins have overlap) in the frequency domain.

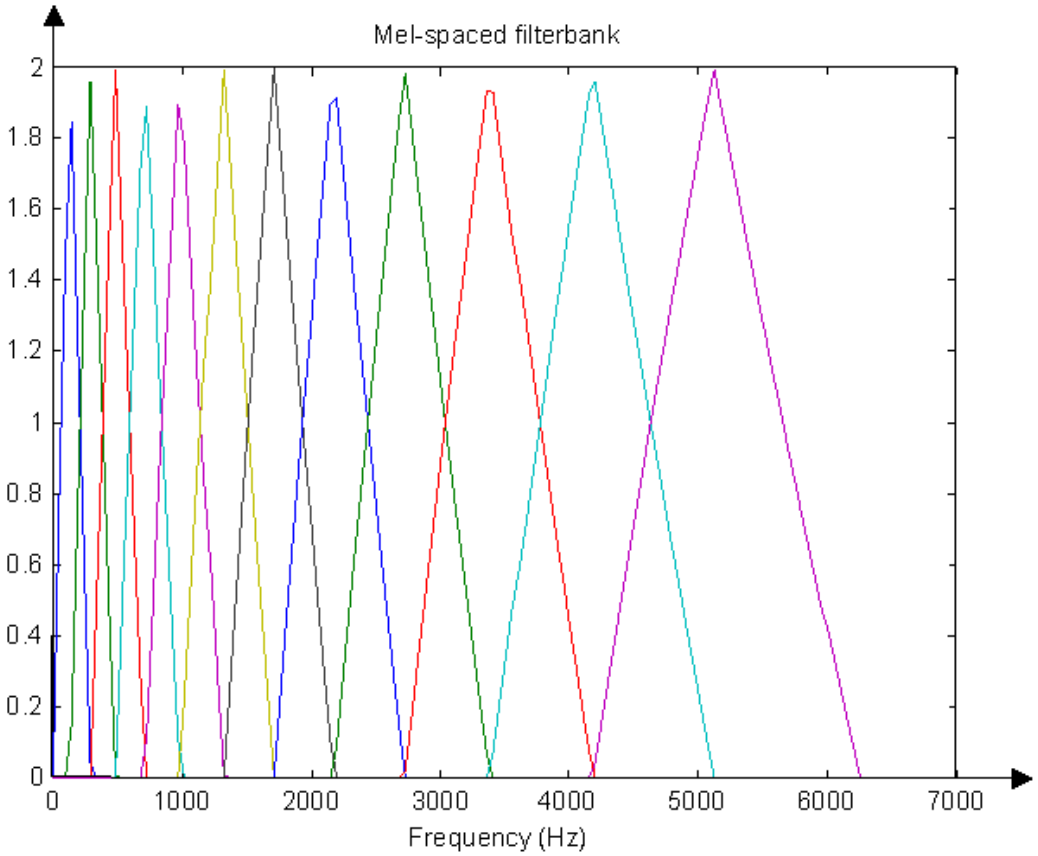


Figure: The Mel spaced filter banks, acting in the frequency domain.

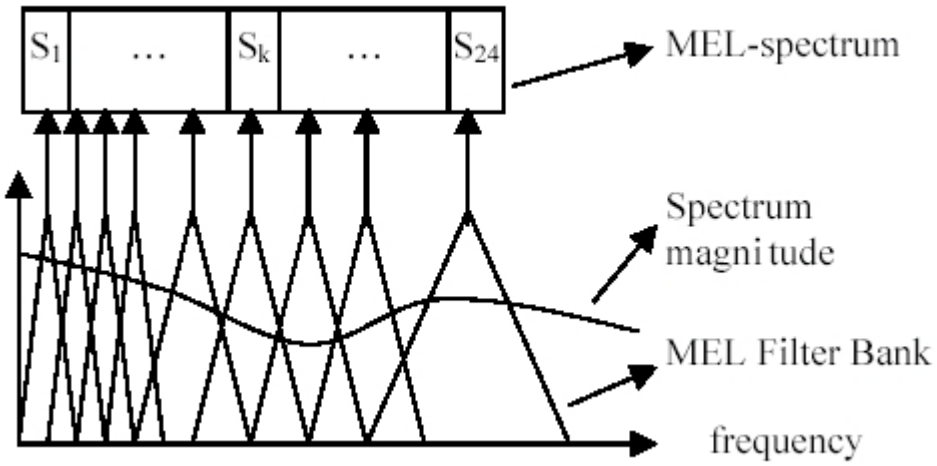


Figure: Mel filter bank operating on a spectrum.

Cepstral Coefficient

The log Mel spectrum is now decorrelated using Discrete Cosine Transform. The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis.

$$y(k) = w(k) \sum_{n=1}^{N} x(n) \cos \frac{\pi(2n-1)(k-1)}{2N}, \qquad k = 1, \ldots, N$$

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 1 \\ \sqrt{\frac{2}{N}}, & 2 \le k \le N \end{cases}$$

y(k) the Cepstral Coefficient is the DCT of Mel spectrum i.e. x(k).

Using DCT we decorrelate the log Mel spectrum of the signal to generate the Mel Frequency Cepstral Coefficients for each frame, thus generating the most important factor for speaker recognition viz. the feature vectors. Each frame contributes a feature vector. The interesting part is the transpose of the feature vectors, which indicates the various discontinuities, repetitions and fundamental frequencies in the speech.

Normalization

The Cepstral coefficient vectors are normalized so that their mean is 0 and variance is 1, to reduce any biasing due to differences in amplitude between other vectors in the analysis stage. For each element x(i) the transform (x(i)- μ) / σ , is carried out, where μ is the mean and σ is the standard deviation of the unnormalized vector.

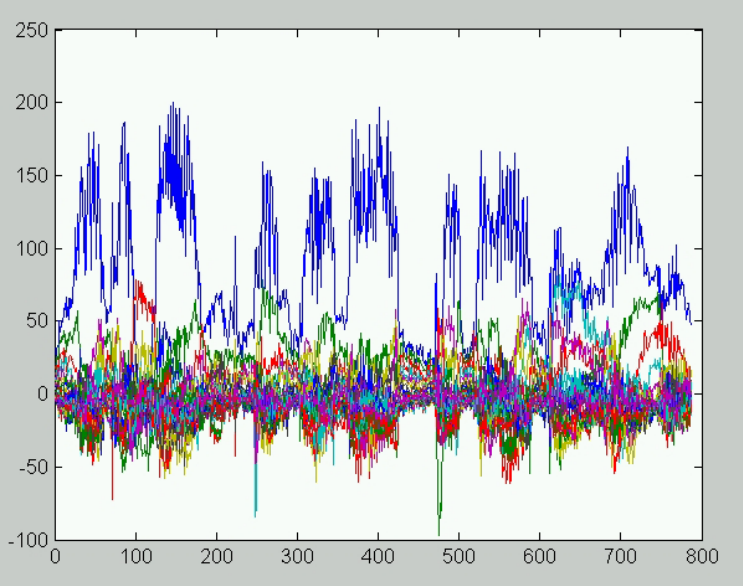
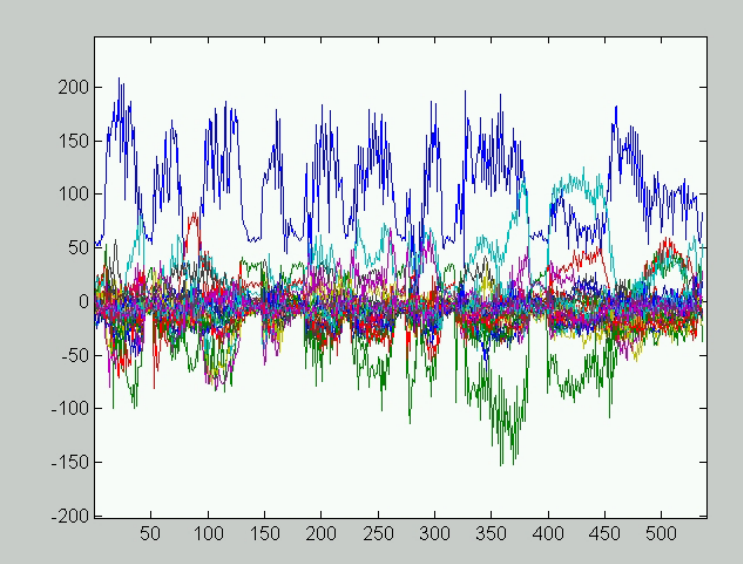


Figure: MFCC of the vocal equivalent of our key sentence, as spoken by an user. Each horizontal row represents a feature vector.



MFCC of the vocal equivalent of our key sentence, spoken by another user. There are perceptible differences in the features.

Crosscorrelation Analysis

Upon generation of the Mel cepstral coefficients, we analyze them for corresponding similarities. Our method of analysis involves the crosscorrelation between each of the n-feature vectors i.e. the Mel cepstrum coefficients of two user voices. The crosscorrelation value between corresponding feature vectors of two voices is calculated.

r(k)=crosscorr(C1(1),C2(1),shift); where cross correlation is given by the formula below.

$$r_{xy}(l) \triangleq \frac{1}{N}(x \star y)(l) \triangleq \frac{1}{N}\sum_{n=0}^{N-1} \overline{x(n)}y(n+l),\ l = 0, 1, 2, \ldots, N-1$$

One vector is kept stationary while the other is shifted up to ±lag limits, and the crosscorrelation value is calculated at each lag value. We take the maximum value of crosscorrelation as the score for two given feature vectors.
C1 and C2 are the cepstral feature vector matrices of two users. Moreover correlation between higher order cepstral vectors is given lesser weight as compared to lower order ones, using a linear gradient. After each run of crosscorrelation, a score is generated, based upon the similarities between the feature vectors. For the case of speaker identification, the system returns the user whose voice samples generated the highest scores, while in the case of speaker verification, the score is tallied against a preset threshold, on exceeding which the user is granted access.

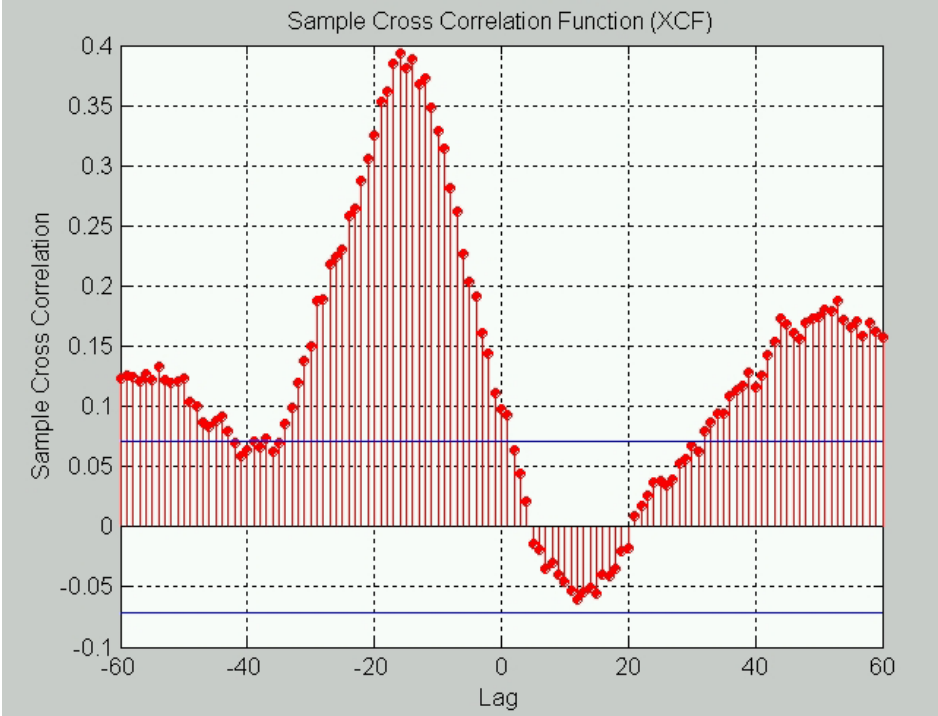


Figure: A crosscorrelation run between feature vectors of the same person with ±60 lag, the score implied is 0.4, at a lag of –16.

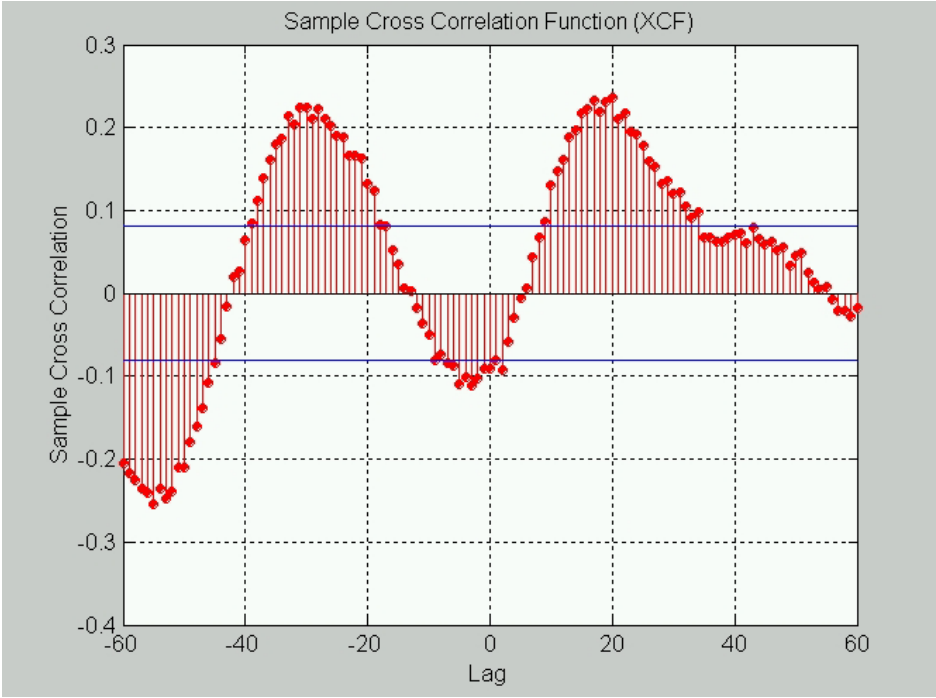


Figure: A crosscorrelation run between feature vectors of two different persons with ±60 lag, the score implied is 0.24, at a lag of +20.

Speaker Recognition System in MATLAB.

MATLAB, the language of technical computing provides us with a plethora of useful functions such as fft, dct, autocorrelation, etc. Moreover handling audio files in **MATLAB** is a breeze with built in functionality to read, record and manipulate **PCM** encoded wave files. The graphics support is excellent resulting is instant analysis of the procedures involved, and fine tuning, thus speeding up the job.

We enumerate below, the steps involved in the Security System.

1) Gathering Feature Vectors from users and maintaining them in a database.
    - Get (userid).
    - Select sample number from 1 to 4.
    - Record the voice sample from user and generate the MFCC (Mel Frequency Cepstral Coefficients).
    - Generate a score based on comparison with previous MFCCs stored.
    - If score > threshold accept sample, else re-record sample.
2) Obtain a voice sample from an user X.
3) Generate the MFCC from the sample.
4) Set the Mode i.e. Speaker Identification / Speaker Verification.
5) If Speaker identification then
    - Compare MFCC of user X against all the stored MFCC, and generate scores for each user in database.
    - Return the user identifier(s), with the best match.
6) If Speaker Verification then
    - Obtain (userid) from user.
    - Compare MFCC of user X against all the stored MFCC of the given user.
    - Return Affirmative if score exceeds threshold, else decline access,

Achievements and preliminary Conclusions

During our preliminary test runs, we loaded up our database with samples from eight students. We recorded some of the samples at different time of the day, to minimize any biases.
A typical test run with eight students indicate the following

1) The system is able to recognize five out of eight students without any major discrepancies.
2) A specific user could neither be recognized nor verified at all,
3) Another user was being confused with a particular user and vice versa.
4) This system is marginally faster than systems based on clustering mechanisms.

Conclusions

1) The system is not very robust, but with the use of good quality microphone and some additional features such as pitch, power spectral density, etc we can improve the robustness where discrepancies arise.

References

1. Alan V Oppenheim, Ronald W. Schafer, "*Digital Signal Processing, Prentice Hall."*
2. Vishwanath Mantha, Richard Duncan, Yufeng Wu, Jie Zhao, Aravind Ganapathiraju, Joseph Picone Institute for Signal and Information Processing Mississippi State University*" IMPLEMENTATION AND ANALYSIS OF SPEECH RECOGNITION FRONTENDS".*
3. Dr. Abbott, Virginia Tech, "*Spoken Word Recognition using Cepstral Analysis"*
4. Minh N. Do, Swiss Federal Institute of Technology, Lausanne, Switzerland*," An Automatic Speaker Recognition System".*
5. TS Chang and Stephen D Hooser. "*Two new methods for Speaker Recognition using Cepstral Analysis."*
6. P Perona, D Psaltis, California Institute of Technology, "S*peaker Recognition System Handout."*
7. Tomi Kinnunen, Ismo Kärkkäinen and Pasi Fränti, University of Joensuu, Department of Computer Science *"IS SPEECH DATA CLUSTERED? - STATISTICAL ANALYSIS OF CEPSTRAL FEATURES."*
8. PravinkumarPremakanthan & Wasfy.B. Mikhael, Fellow IEEE, University of Central Florida, Orlando*, "SPEECH FEATURE EXTRACTION SPEAKER VERIFICATION/RECOGNITION AND THE IMPORTANCE OF SELECTIVE FEATURE EXTRACTION: REVIEW."*
9. *"Biometrics research", http://biometrics.cse.msu.edu/index.html.*
10. N. Johan Wismer, Brüel& Kjær, Denmark, "*Gearbox Analysis using Cepstrum Analysis and Comb Liftering."*