VOICE RECOGNITION SYSTEM S

NIKHIL K. SINGH

This project basically involved understanding the existing techniques of isolated word recognition, simulating the work using the available tool MATLAB

ROLL NO. 1405231028

B.TECH 3RD YEAR

ELECTRONICS AND COMMUNICATION

INSTITUE OF ENGINEERING AND TECHNOLOGY

SUBMITTED ON:

INTRODUCTION

The concept of a machine than can recognize the human voice has long been an accepted feature in Science Fiction. From “Star Trek‟ to George Orwell’s “1984‟ - “Actually he was not used to writing by hand. Apart from very short notes, it was usual to dictate everything into the speakwriter.” - It has been commonly assumed that one day it will be possible to converse naturally with an advanced computer-based system.

Indeed in his book “The Road Ahead‟, Bill Gates hails “**Automatic Speaker Recognition (ASR)”** as one of the most important innovations for future computer operating systems.

From a technological perspective it is possible to distinguish between two broad types of ASR:

- Direct voice input (DVI)
- Large vocabulary continuous speech recognition (LVCSR).

DVI devices are primarily aimed at voice command-and control, whereas LVCSR systems are used for form filling or voice-based document creation.

DVI systems are typically configured for small to medium sized vocabularies (up to several thousand words) and might employ word or phrase spotting techniques. Also, DVI systems are usually required to respond immediately to a voice command.

LVCSR systems involve vocabularies of perhaps hundreds of thousands of words, and are typically configured to transcribe continuous speech.

MOTIVATION FOR ASR

The motivation for ASR is simple: it is man’s principle means of communication and is, therefore, a convenient and desirable mode of communication with machines.

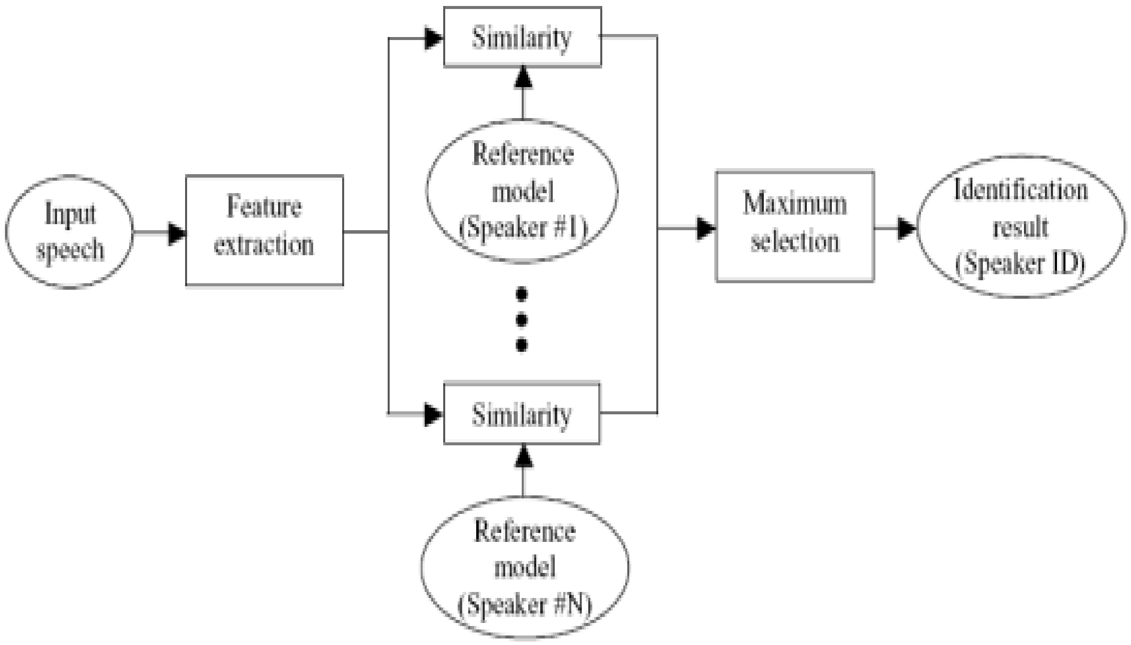
Speech communication has evolved to be efficient and robust and it is clear that the route to computer based speech recognition is the modeling of the human system. Unfortunately from pattern recognition point of view - humans recognize speech through a very complex interaction between many levels of processing:

- Using syntactic and semantic information
- Powerful low level pattern classification and processing.

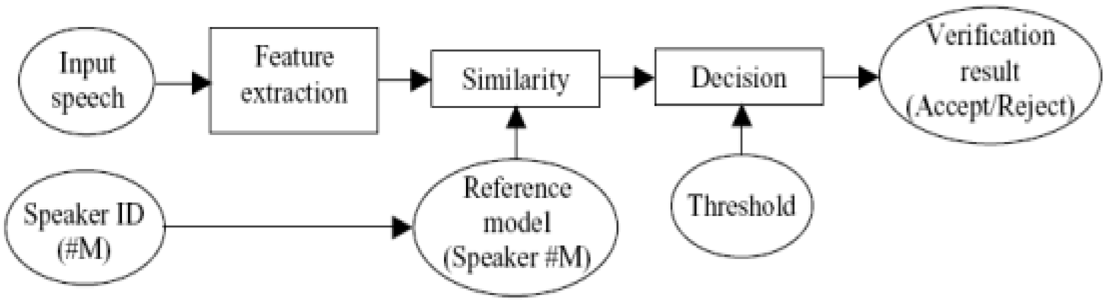
Automatic speech recognition is therefore an engineering compromise between the ideal i.e. a complete model of the human, and the practical i.e. the tools that science and technology provide and that costs allow.

At the highest level, all speaker recognition systems contain two main modules feature extraction and feature matching. Feature extraction is the process that extracts a small amount of data from the voice signal that can later be used to represent each speaker. Feature matching involves the actual procedure to identify the unknown speaker by comparing extracted features from his/her voice input with the ones from a set of known speakers. We will discuss each module in detail in later sections.

All Recognition systems have to serve two different phases. The first one is referred to the enrollment sessions or training phase while the second one is referred to as the operation sessions or testing phase. In the training phase, each registered speaker has to provide samples of their speech so that the system can build or train a reference model for that speaker. In case of speaker verification systems, in addition, a speaker-specific threshold is also computed from the training samples. During the testing (operational) phase, the input speech is matched with stored reference model(s) and recognition decision is made.


Input speech
Feature extraction
Similarity
Reference model (Speaker #1)
Similarity
Reference model (Speaker #N)
Maximum selection
Identification result (Speaker ID)

Automatic speech recognition works based on the premise that a person‟s speech exhibits characteristics that are unique to the speaker. However this task has been challenged by the highly variant of input speech signals. The principle source of variance is the speaker himself. Speech signals in training and testing sessions can be greatly different due to many facts such as people voice change with time, health conditions (e.g. the speaker has a cold), speaking rates, etc. There are also other factors, beyond speaker variability, that present a challenge to speech recognition technology. Examples of these are acoustical noise and variations in recording environments (e.g. speaker uses different telephone handsets). The challenge would be make the system “Robust”.



ASR SYSTEM

INTRODUCTION

Speech processing is the study of speech signals and the processing methods of these signals. The signals are usually processed in a digital representation whereby speech processing can be seen as the interaction of digital signal processing and natural language processing.

Speech coding:

It is the compression of speech (into a code) for transmission with speech codecs that use audio signal processing and speech processing techniques. The techniques used are similar to that in audio data compression and audio coding where knowledge in psychoacoustics is used to transmit only data that is relevant to the human auditory system. For example, in narrow band speech coding, only information in the frequency band of 400 Hz to 3500 Hz is transmitted but the reconstructed signal is still adequate for intelligibility.

Speech coding differs from audio coding in that there is a lot more statistical information available about the properties of speech. In addition, some auditory information which is relevant in audio coding can be unnecessary in the speech coding context.

In speech coding, the most important criterion is preservation of intelligibility and "pleasantness" of speech, with a constrained amount of transmitted data. should be emphasized that the intelligibility of speech includes, besides the actual literal content, also speaker identity, emotions, intonation, timbre etc. that are all important for perfect intelligibility.

The more abstract concept of pleasantness of degraded speech is a different property than intelligibility, since it is possible that degraded speech is completely intelligible, but subjectively annoying to the listener.

Speech synthesis:

Speech synthesis is the artificial production of human speech. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can also be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity.

For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

Voice analysis:

Voice problems that require voice analysis most commonly originate from the vocal cords since it is the sound source and is thus most actively subject to tiring. However, analysis of the vocal cords is physically difficult. The location of the vocal cords effectively prohibits direct measurement of movement. Imaging methods such as x-rays or ultrasounds do not work because the vocal cords are surrounded by cartilage which distorts image quality.

Most important indirect methods are inverse filtering of sound recordings and electroglottographs (EGG). In inverse filtering methods, the speech sound is recorded outside the mouth and then filtered by a mathematical method to remove the effects of the vocal tract.

This method produces an estimate of the waveform of the pressure pulse which again inversely indicates the movements of the vocal cords. The other kind of inverse indication is the electroglottographs, which operates with electrodes attached to the subject‟s throat close to the vocal cords. Changes in conductivity of the throat 15 indicate inversely how large a portion of the vocal cords are touching each other.

It thus yields one-dimensional information of the contact area. Neither inverse filtering nor EGG is thus sufficient to completely describe the glottal movement and provide only indirect evidence of that movement.

Speech recognition:

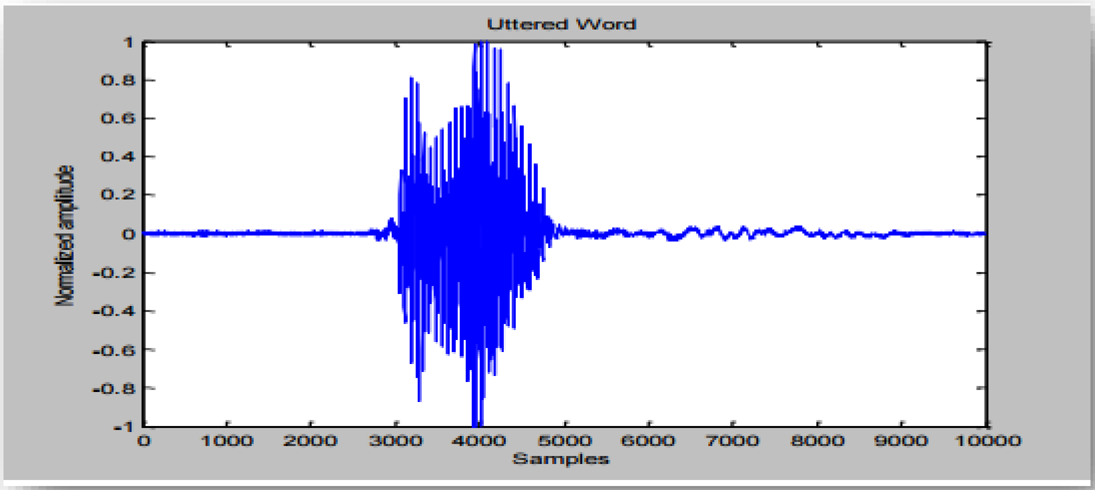
Speech recognition is the process by which a computer (or other type of machine) identifies spoken words. Basically, it means talking to your computer, and having it correctly recognize what you are saying.

This is the key to any speech related application. As shall be explained later, there are a number ways to do this but the basic principle is to somehow extract certain key features from the uttered speech and then treat those features as the key to recognizing the word when it is uttered again.

SPEECH RECOGNITION BASICS

Utterance

An utterance is the vocalization (speaking) of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences. 0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000 -1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1 Samples Normalized Utterance of "HELLO"



Speaker Dependence

Speaker dependent systems are designed around a specific speaker. They generally are more accurate for the correct speaker, but much less accurate for other speakers. They assume the speaker will speak in a consistent voice and tempo. Speaker independent systems are designed for a variety of speakers. Adaptive systems usually 16 start as speaker independent systems and utilize training techniques to adapt to the speaker to increase their recognition accuracy.

Vocabularies

Vocabularies (or dictionaries) are lists of words or utterances that can be recognized by the SR system. Generally, smaller vocabularies are easier for a computer to recognize, while larger vocabularies are more difficult. Unlike normal dictionaries, each entry doesn't have to be a single word. They can be as long as a sentence or two. Smaller vocabularies can have as few as 1 or 2 recognized utterances (e.g.” Wake Up"), while very large vocabularies can have a hundred thousand or more!

Accuracy

The ability of a recognizer can be examined by measuring its accuracy - or how well it recognizes utterances. This includes not only correctly identifying an utterance but also identifying if the spoken utterance is not in its vocabulary. Good ASR systems have an accuracy of 98% or more! The acceptable accuracy of a system really depends on the application.

Training

Some speech recognizers have the ability to adapt to a speaker. An ASR system is trained by having the speaker repeat standard or common phrases and adjusting its comparison algorithms to match that particular speaker. Training a recognizer usually improves its accuracy. As long as the speaker can consistently repeat an utterance, ASR systems with training should be able to adapt.

CLASSIFICATION OF ASR SYSTEM

A speech recognition system can operate in many different conditions such as speaker dependent/independent, isolated/continuous speech recognition, for small/large vocabulary. Speech recognition systems can be separated in several different classes by describing what types of utterances they have the ability to recognize. These classes are based on the fact that one of the difficulties of ASR is the ability to determine when a speaker starts and finishes an utterance.

- ISOLATED WORDS
- CONNECTED WORDS
- CONTINUOUS SPEECH
- SPONTANEOUS SPEECH
- SPEAKER DEPENDENCE

Isolated word recognizers usually require each utterance to have quiet (lack of an audio signal) on BOTH sides of the sample window. It doesn't mean that it accepts single words, but does require a single utterance at a time.

Connect word systems (or more correctly 'connected utterances') are similar to Isolated words, but allow separate utterances to be 'run-together' with a minimal pause between them.

Continuous recognition is the next step. Recognizers with continuous speech capabilities are some of the most difficult to create because they must utilize special methods to determine utterance boundaries. Continuous speech recognizers allow users to speak almost naturally, while the computer determines the content.

Spontaneous speech can be thought of as speech that is natural sounding and not rehearsed. An ASR system with spontaneous speech ability should be able to 18 handle a variety of natural speech features.

Speaker Dependent systems are trained with one speaker and recognition is done only for that speaker. Speaker Independent systems are trained with one set of speakers. This is obviously much more complex than speaker dependent recognition. A problem of intermediate complexity would be to train with a group of speakers and recognize speech of a speaker within that group.

DIFFICULTIES IN AUTOMATIC SPEECH RECOGNITION

There are a few problems in speech recognition that haven‟t yet been discovered. However there are a number of problems that have been identified over the past few decades most of which still remain unsolved.

Some of the main problems in ASR are:

1. Determining word boundaries Speech is usually continuous in nature and word boundaries are not clearly defined. This happens when the speaker is speaking at a high speed. Varying Accents People from different parts of the world pronounce words differently.

2. Large vocabularies: When the number of words in the database is large, similar sounding words tend to cause a high amount of error i.e. there is a good probability that one word is recognized as the other.

3. Changing Room Acoustics Noise is a major factor in ASR. In fact it is in noisy conditions or in changing room acoustic that the limitations of present day ASR engines become prominent.

4. Temporal Variance Different speakers speak at different speeds. Present day ASR engines just cannot adapt to that.

SPEECH ANALYZER

Speech analysis: also referred to as front-end analysis or feature extraction, is the first step in an automatic speech recognition system.

This process aims to extract acoustic features from the speech waveform.

The output of front-end analysis is a compact, efficient set of parameters that represent the acoustic properties observed from input speech signals, for subsequent utilization by acoustic modelling.

There are three major types of front-end processing techniques:

1. Linear predictive coding (LPC)
2. Mel-frequency cepstral coefficients (MFCC)
3. Perceptual linear prediction (PLP)

where the latter two are most commonly used in state-of-the-art ASR systems.

Linear predictive coding (LPC) starts with the assumption that a speech signal is produced by a buzzer at the end of a tube (voiced sounds), with occasional added hissing and popping sounds. Although apparently crude, this model is actually a close approximation to the reality of speech production.

Mel Frequency Cepstrum Coefficients

These 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 or DCT.

This can allow for better processing of data, for example, in audio compression. However, unlike the sonogram, MFCCs lack an outer ear model and, hence, cannot represent perceived loudness accurately.

Perceptual Linear Prediction

Perceptual linear prediction, similar to LPC analysis, is based on the short-term spectrum of speech. In contrast to pure linear predictive analysis of speech, perceptual linear prediction (PLP) modifies the short-term spectrum of the speech by several psychophysically based transformations.

This technique uses three concepts from the psychophysics of hearing to derive an estimate of the auditory spectrum:

(1) The critical-band spectral resolution

(2) The equal-loudness curve

(3) The intensity-loudness power law.

The auditory spectrum is then approximated by an autoregressive all-pole model. In comparison with conventional linear predictive (LP) analysis, PLP analysis is more consistent with human hearing.

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.

SPEECH CLASSIFIER

The problem of ASR belongs to a much broader topic in scientific and engineering so called pattern recognition.

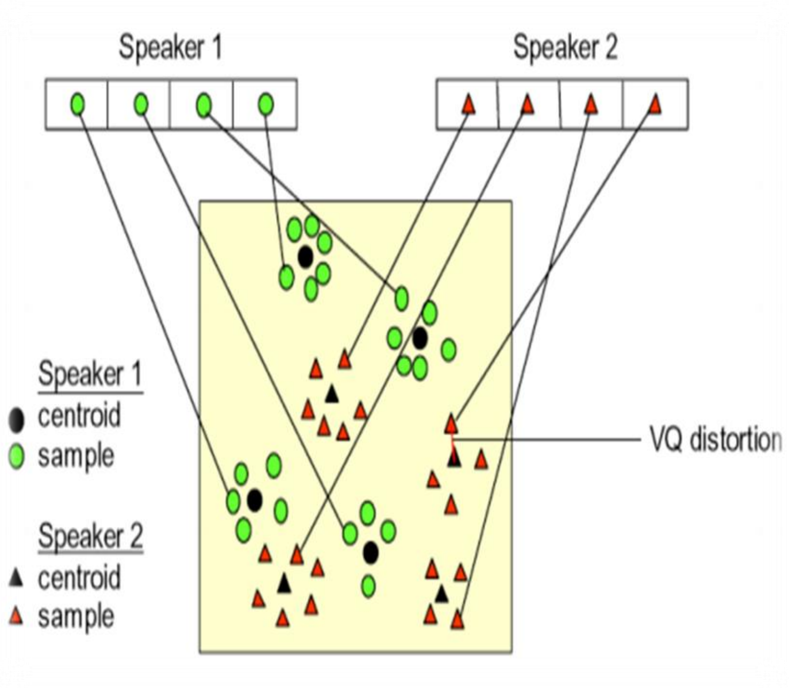
The goal of pattern recognition is to classify objects of interest into one of a number of categories or classes. The objects of interest are generically called patterns and in our case are sequences of acoustic vectors that are extracted from an input speech using the techniques described in the previous section.

The classes here refer to individual speakers. Since the classification procedure in our case is applied on extracted features, it can be also referred to as feature matching.

Feature matching techniques used in speaker recognition include:

- Dynamic Time Warping (DTW).
- Hidden Markov Modeling (HMM).
- Vector Quantization (VQ).
- *Dynamic time warping is an algorithm for measuring similarity between two sequences which may vary in time or speed.*
- *For instance, similarities in walking patterns would be detected, even if in one video the person was walking slowly and if in another they were walking more quickly, or even if there were accelerations and decelerations during the course of one observation.*
- *DTW has been applied to video, audio, and graphics -indeed, any data which can be turned into a linear representation can be analyzed with DTW.*
- *Hidden Markov Model characterizes words into probabilistic models wherein the various phonemes which contribute to the word represent the states of the HMM while the transition probabilities would be the probability of the next phoneme being uttered (ideally 1.0).*

- *Models for the words which are part of the vocabulary are created in the training phase.*
- *Now, in the recognition phase when the user utters a word it is split up into phonemes as done before and it's HMM is created. After the utterance of a particular phoneme, the most probable phoneme to follow is found from the models which had been created by comparing it with the newly formed model.*
- *Such a probabilistic system would be more efficient than just cepstral analysis as these is some amount of flexibility in terms of how the words are uttered by the users.*
- *__Vector Quantization (VQ)__ is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and can be represented by its center called a centroid. The collection of all code words is called a codebook.*



In the figure, only two speakers and two dimensions of the acoustic space are shown. The circles refer to the acoustic vectors from the speaker 1 while the triangles are from the speaker 2. In the training phase, a speaker-specific VQ codebook is generated for each known speaker by clustering his/her training acoustic vectors.

The result code words (centroids) are shown in by black circles and black triangles for speaker 1 and 2, respectively.

The distance from a vector to the closest codeword of a codebook is called a VQ-distortion.

In the recognition phase, an input utterance of an unknown voice is “vector-quantized” using each trained codebook and the total VQ distortion is computed. The speaker corresponding to the VQ codebook with smallest total distortion is identified.

FEATURE EXTRACTION

PREPROCESSING

First the input speech signal is pre-emphasized to artificially amplify the high frequencies. Speech signals are non-stationary signals, meaning to say the transfer function of the vocal tract; which generates it, changes with time, though it changes gradually.

It is safe to assume that it is piecewise stationary. Thus the speech signal is framed with 30ms to 32ms frames with an overlap of 20ms. The need for the overlap is that information may be lost at the frame boundaries, so frame boundaries need to be within another frame.

Mathematically, framing is equivalent to multiplying the signal with a series of sliding rectangular windows. The problem with rectangular windows is that the power contained in the side lobes is significantly high and therefore may give rise to spectral leakage.

In order to avoid this we use a Hamming window given by:

$$\boldsymbol{w(n) = 0.54 - 0.46\cos\left({}^{2\pi n}/_{N-1}\right) \qquad 0 \le n \le N-1}$$

MEL FREQUENCY CEPSTRAL COEFFICIENTS MFCCs are typically computed by using a bank of triangular-shaped filters, with the center frequency of the filter spaced linearly for frequencies less than 1000 Hz and logarithmically above 1000 Hz.

The bandwidth of each filter is determined by the center frequencies of the two adjacent filters and is dependent on the frequency range of the filter bank and number of filters chosen for design. But for the human auditory system it is estimated that the filters have a bandwidth that is related to the center frequency of the filter. Further it has been shown that there is no evidence of two regions (linear and logarithmic) in the experimentally determined Mel frequency scale.

Silence detection

This is a very important step in any front end speaker recognition system. When the user says out any word the system will never have control over the instant the word is uttered. It is for this reason that we need a silence detection step which basically determines when the user has actually started uttering the word any thereby translates the frame of reference to that instant.

Windowing

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of the frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of the frame.

$$\boldsymbol{w(n), 0 \le n \le N-1}$$

N: no. of samples in each frame

Then the result of windowing is the signal :

Y 1 (n) = x_1(n)w(n) , $0 \le n \le N-1$

Typically the hamming window is used which has the form:

$$\mathbf{w(n) = 0.54 - 0.46\cos(^{2\pi n}/_{N-1})} \qquad \mathbf{0 \le n \le N-1}$$

Fast Fourier Transform (FFT)

The next processing step is the Fast Fourier Transform, which converts each frame of N samples from the time domain into the frequency domain. The FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT) which is defined on the set of N samples { Xn }, as follows:

$$\mathrm{Xn} = \sum_{k=0}^{N-1} x_k e^{-2jkn\pi/N} \qquad n = 0,1,2.....N-1$$

Note that we use j here to denote the imaginary unit. In general Xn's are complex numbers. The resulting sequence {Xn} is interpreted as follows: the zero frequency corresponds to n = 0, positive frequencies: 0 < f < Fs/2 correspond to values N/2-1 while negative frequencies –Fs/2≤n≤1 < f < N-1. ≤n ≤0 correspond to N/2+1

Here, F_s denotes the sampling frequency. The result obtained after this step is often referred to as signal's spectrum or periodogram

Mel-frequency Warping

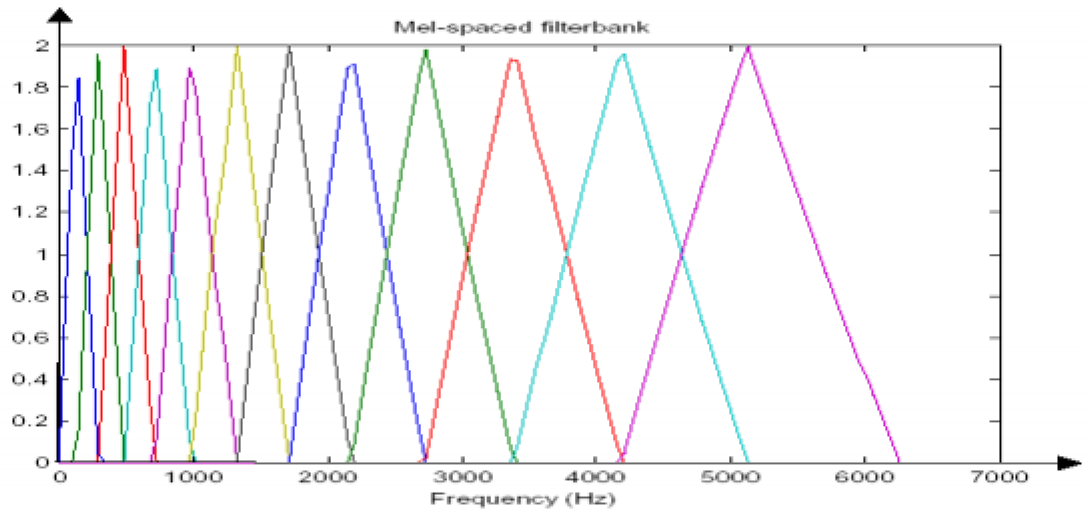
Psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency, f, measured in Hz, a subjective pitch is measured on a scale called the “Mel” scale. The Mel-frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. 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:

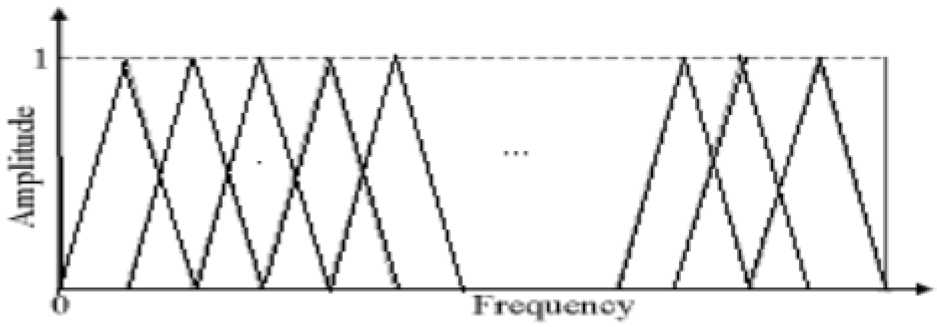
$$mel(f) = 2595 * log_{10}(1 + f/700)$$

One approach to simulating the subjective spectrum is to use a filter bank, one filter for each desired mel-frequency component. That filter bank has a triangular bandpass frequency response, and the spacing as well as the bandwidth is determined) thus consists of$\omega$by a constant mel-frequency interval.

The modified spectrum of S ( ) is the input. The number of mel cepstral$\omega$the output power of these filters when S (coefficients, K, is typically chosen as 20. Note that this filter bank is applied in the frequency domain; therefore it simply amounts to taking those triangle-shape windows.

A useful way of thinking about this mel-warped filter bank is to view each filter as a histogram bin (where bins have overlap) in the frequency domain. A useful and efficient way of implementing this is to consider these triangular filters in the Mel scale where they would in effect be equally spaced filters.





[Filterbank in linear freq. scale and mel freq. scale]

Cepstrum

In this final step, we convert the log Mel spectrum back to time.

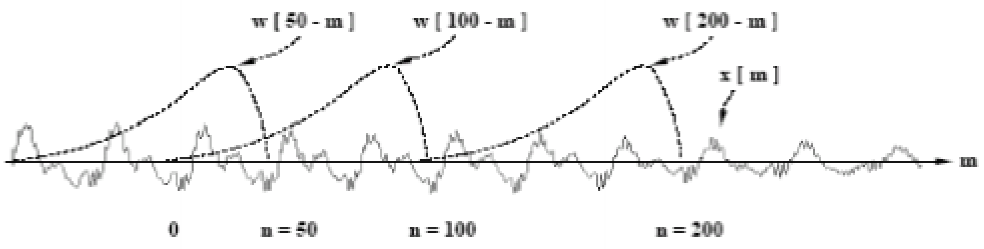
The result is called the mel frequency cepstral coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis.

Because the mel spectrum coefficients (and so their logarithm) are real numbers, we can convert them to the time domain using the Discrete Cosine Transform (DCT). Therefore if we denote those mel power spectrum coefficients that are the result of the last step are Sk , where k =1, 2,.....,K. Calculate the MFCC's c_n as :

$$\widetilde{c_n} = \sum_{k=1}^{k} (log\widetilde{s_k})cos[n(k - {}^{1}/_{2})\frac{\pi}{k}]$$

TIME DOMAIN ANALYSIS

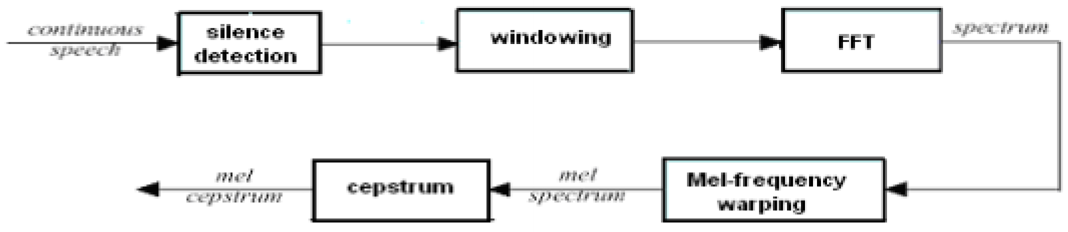
Although speech is non-stationary in a true sense, study has shown that it can be assumed to be quasi-stationary i.e. slowly time varying. When examined over a sufficiently short period of time (between 20 and 40 msec), its characteristics are fairly stationary. The implication of this is that when we consider such small segments of speech they have a dominant frequency within the windowed segment. Therefore, this can be processed through a short-time frequency analysis. The figure below shows how we can use overlapping time domain windows so as to cater for the short time Fourier analysis.



[overlapping windows]

MFCC APPROACH:

The speech input is typically recorded at a sampling rate above 10000 Hz. This sampling frequency was chosen to minimize the effects of aliasing in the analog-todigital conversion. These sampled signals can capture all frequencies up to 5 kHz, which cover most energy of sounds that are generated by humans. The main purpose of the MFCC processor is to mimic the behavior of the human ears. In addition, rather than the speech waveforms themselves, MFCC‟s are shown to be less susceptible to mentioned variations.



[MFCC PROCESSOR]

- We first stored the speech signal as a 10000 sample vector. It was observed from our experiment that the actual uttered speech eliminating the static portions came up to about 2500 samples, so, by using a simple threshold technique we carried out the silence detection to extract the actual uttered speech.

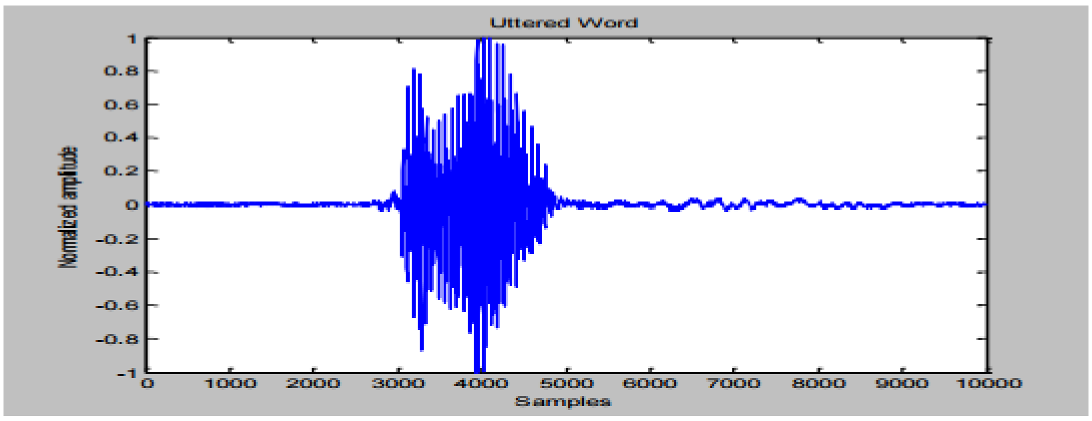
- It is clear that what we wanted to achieve was a voice based biometric system capable of recognizing isolated words. As our experiments revealed almost all the isolated words were uttered within 2500 samples. But, when we passed this speech signal through a MFCC processor, it spilt this up in the time domain by using overlapping windows each with about 250 samples.

- When we convert this into the frequency domain we just have about 250 spectrum values under each window. This implied that converting it to the Mel scale would be redundant as the Mel scale is linear till 1000 Hz. So, we eliminated the block which did the Mel warping.

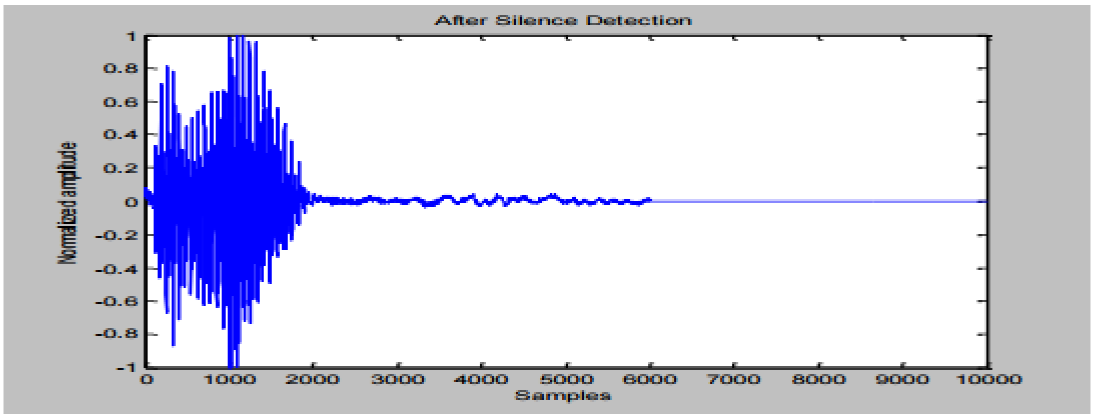
- We directly used the overlapping triangular windows in the frequency domain. We obtained the energy within each triangular window, followed by the DCT of their logarithms to achieve good compaction within a small number of coefficients as described by the MFCC approach.

- This algorithm however, has a drawback. As explained earlier the key to this approach is using the energies within each triangular window, however, this may not be the best approach as was discovered. It was seen from the experiments that because of the prominence given to energy, this approach failed to recognize the same word uttered with different energy.

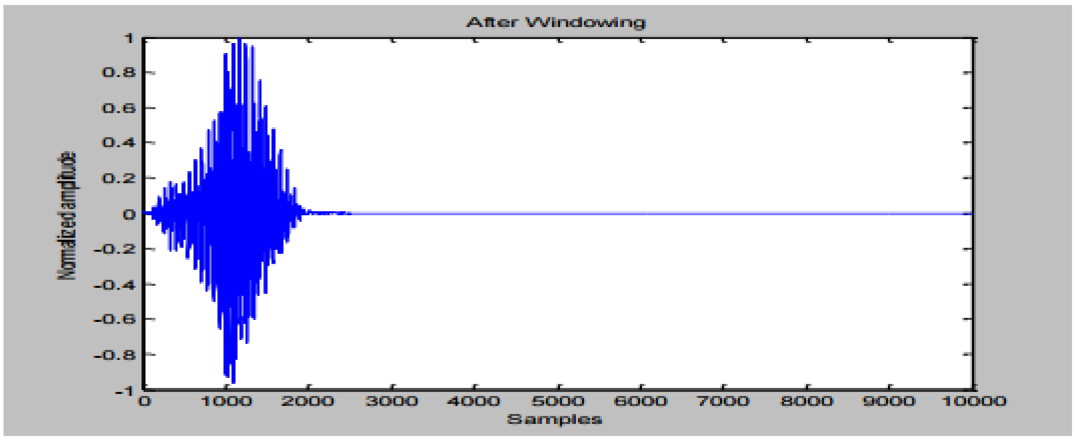
- As this takes the summation of the energy within each triangular window it would essentially give the same value of energy irrespective of whether the spectrum peaks at one particular frequency and falls to lower values around it or whether it has an equal spread within the window. This is why we decided not to go ahead with the implementation of the MFCC approach. The simulation was carried out in MATLAB. The various stages of the simulation have been represented in the form of the plots shown. The input continuous speech signal considered as an example for this project is the word “HELLO”.



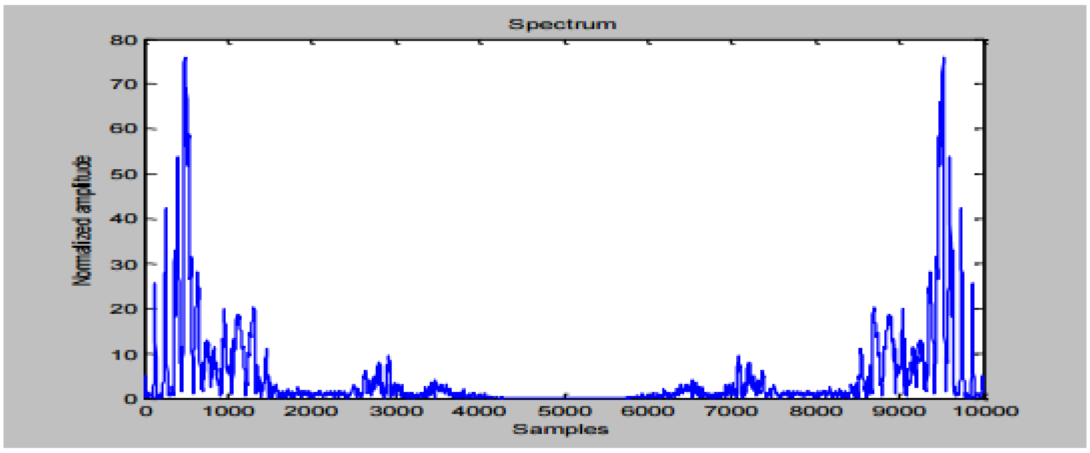
HELLO SAMPLE



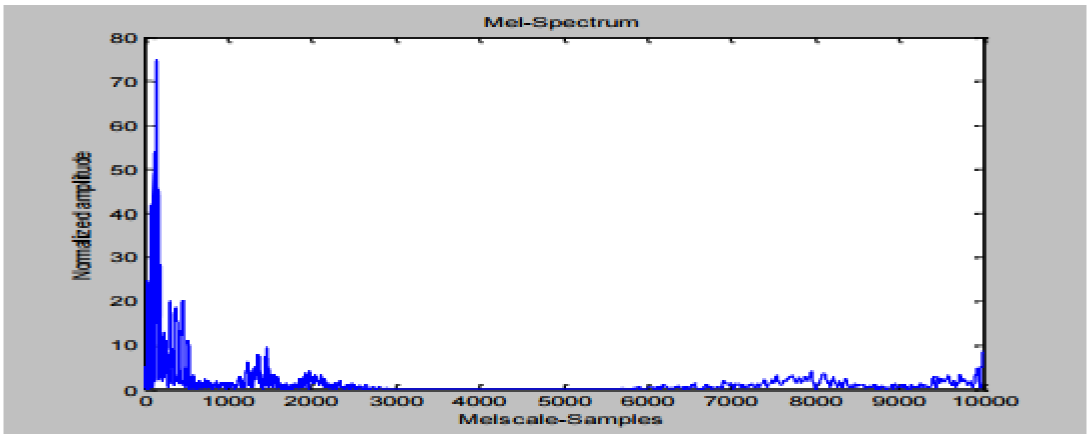
SILENCE DEDUCTION



WINDOWED

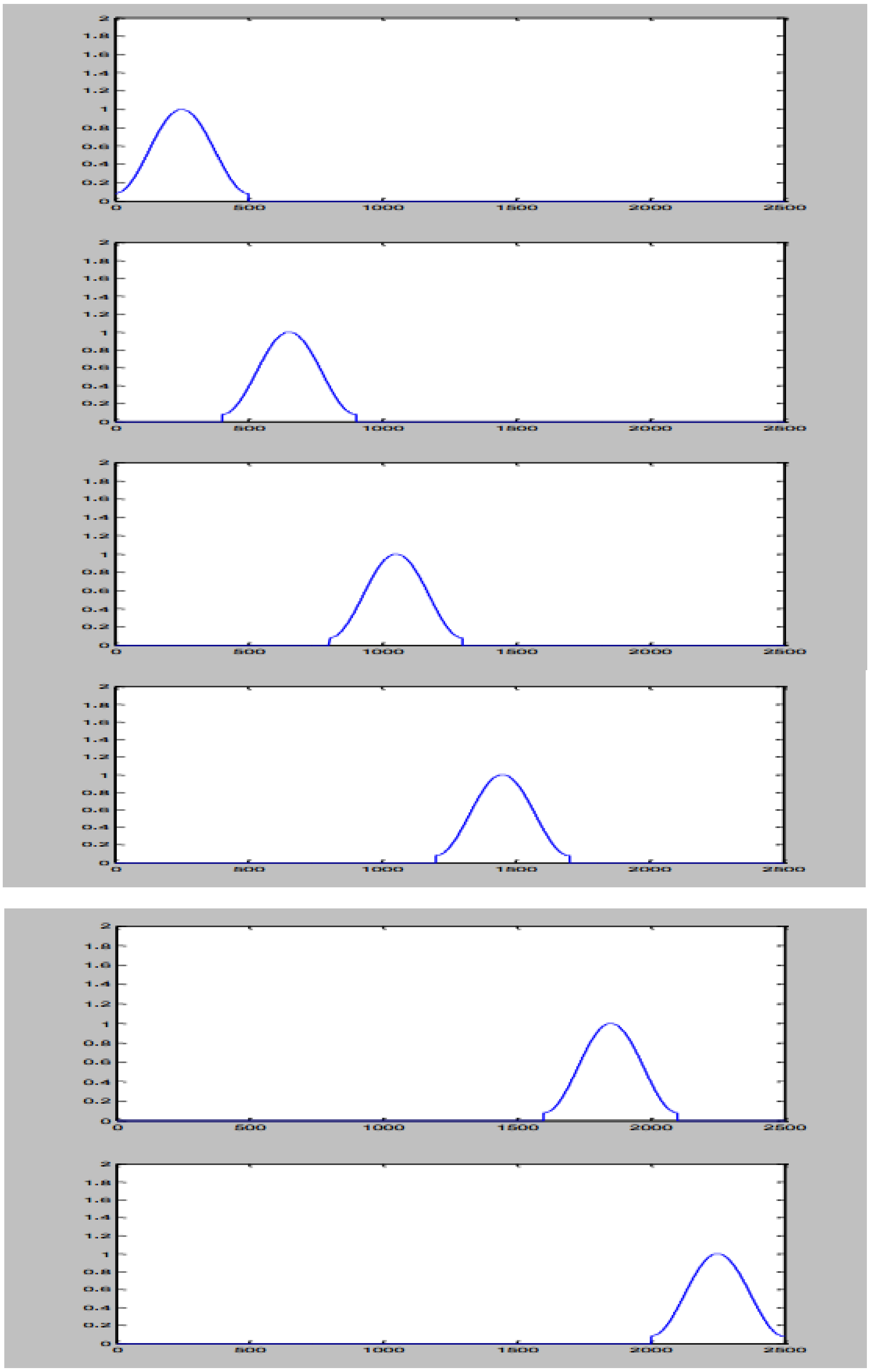


PRE PROCESSED

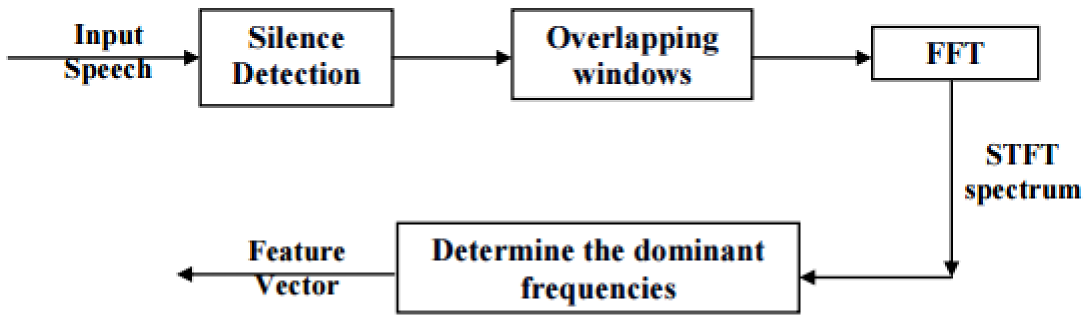


MEL-WRAPPING

TIME DOMAIN APPROACH



OVERLAPPING TIME DOMAIN HAMMING WINDOWS



BLOCK DIAGRAM OF TIME DOMAIN APPROACH

ASR ALGORITHM OPTIMIZATION

INTRODUCTION

Despite the continual growth of the computing power of DSP, direct implementation of ASR algorithms can't over real-time processing.

When porting speech recognition algorithms to embedded platforms, the essential procedures will be included:

Analyze the algorithm and identify the computationally most critical parts.

Simplify the computation by, e.g. using look-up table, avoiding complex¬ function, etc. Avoid unnecessarily repetitive computation.

Pre-compute frequently used parameters as pre-stored constant values¬ whenever possible. Use low-level language if appropriate.

In this work, we have used a set of optimizations for ASR algorithms. By analyzing each computation part in ASR, the most effective optimization methods are applied. Feature extraction techniques Feature extraction is computationally intensive because it involves many signal processing steps such as windowing, Fast Fourier Transform, Mel-scale filter banks, Discrete Cosine transform and so on. For each of these steps, different approximation techniques have been considered. Windowing In this work, we use the Hamming window .

For a frame with fixed-length N, the window shape is fixed and it is not needed to calculate the window function h (n) for each incoming frame. Instead, h (n) ; 0 < N is pre-stored in the RAM. In this way, we save N cosine operations, N multiplications and N additions which require a lot of CPU cycles. We will need N 16-bit words of RAM to store the window function which is not overburden compared to the execution time it costs.

FFT Implementation Although the Fast Fourier Transform (FFT) is much more efficient than the original Discrete Fourier Transform (DFT), it still requires the computation of many exponential or trigonometric functions and multiplications that would take a long execution time on DSPs. In our work, we have used a 256 point DIF –Radix 2 FFT. Since speech is a real-valued signal, each input frame contains 256 real elements .The result contains the first half i.e. 128 complex elements as the FFT output. This output still provides the full information because an FFT of a real sequence has even symmetry around the center or Nyquist point (N/2)

MEMORY REQUIREMENTS CONSIDERATION

Embedded DSP system has limited memory resource. For the front-end part, the major memory requirement comes from storage of look-up table for DCT and gains of filterbanks, and some pre-computed parameters such as the weights of Hamming windows, the center frequencies of each filter bank and cepstral weights. In our work, memory requirements arose from the fact that the speech samples (10000 samples /sec) need to be initially stored, windowed and suitably processed. The default space selected by CCS corresponds to IRAM which doesn‟t satisfy the memory storage requirements. So the SDRAM space was made use to store the parameters. Suitable heap structures were defined in SDRAM and the parameters dumped into these.

Typical these included:

- Input speech samples > SDRAM
- Hamming window points > Implemented as a lookup table and stored in IRAM
- FFT and Magnitude of FFT > IRAM
- In this case only the Windowed signal > SDRAM

It is not advisable to evaluate FFT using the twiddle factors and the data in the SDRAM, so, we created a buffer space in the IRAM where the input samples were fetched from external memory, stored and the FFT was evaluated.

This avoids multiple external memory accesses and speeds up the process.

The feature vectors evaluated after the training phase would be required for the testing phase. It was seen that the data in the IRAM was not overwritten provided we declared a proper section for all our required data; so, we stored the feature vector in the predefined section in the IRAM thus retaining the data even for the testing phase.

Controlling Of Device Through Voice Recognition Using MATLAB

The technique described is one in which firstly a speech command can be determined by power of speech signal which can be taken by the help of microphones being connected to the computer itself.

Using MATLAB Programming the sampling of the speech signal takes place with sampling rate of 8000 samples/sec according to nyquist criteria i.e. F=2*fm Where F=Sampling Frequency, fm = maximum component of frequency being present in speech signal. The sampled signal is then filtered off by using band pass filter lying in the range of 300 Hz-4000 Hz, which filters all the speech signal lying below 300 Hz of frequency range.

Moreover, it includes the algorithm for the creation of speech templates which can be achieved by calculating the power of each sampled signals respectively. Now, a dictionary is created in which same command is taken many times and thus the average of these will represent a speech template which can be then stored in a library.

Now the command templates that are being received by microphone are than compared with the templates being stored in library according to Euclidean`s Distance i.e. Euclidian Distance=Σi =1((dic[i]- com[i])2) Where i denotes the number of sample points. Thus, the command will be detected by a particular device and it will performs the operation accordingly & if it does not match, the device should not follow that command. The general block diagram of the interfacing is shown in fig.6

APPLICATIONS

After nearly sixty years of research, speech recognition technology has reached a relatively high level. However, most state-of-the-art ASR systems run on desktop with powerful microprocessors, ample memory and an ever-present power supply.

In these years, with the rapid evolvement of hardware and software technologies, ASR has become more and more expedient as an alternative human-to-machine interface that is needed for the following application areas:

Stand-alone consumer devices such as wrist watch, toys and hands-free¬ mobile phone in car where people are unable to use other interfaces or big input platforms like keyboards are not available.

Single purpose command and control system such as voice dialing for - cellular, home, and office phones where multi-function computers (PCs) are redundant.

Some of the applications of speaker verification systems are:

- Time and Attendance Systems
- Access Control Systems
- Telephone-Banking/Broking
- Biometric Login to telephone aided shopping systems
- Information and Reservation Services
- Security control for confidential information
- Forensic purposes

Eg:-telephone dialing--->The key focus of this application is to aid the physically challenged in executing a mundane task like telephone dialing. Here the user initially trains the system by uttering the digits from 0 to 9. Once the system has been trained, the system can recognize the digits uttered by the user who trained the system. This system can also add some inherent security as the system based on cepstral approach is speaker dependent. The algorithm is run on a particular speaker and the MFCC coefficients determined. Now the algorithm is applied to a different speaker and the mismatch was clearly observed. Typically a user has two levels of check. He/She has to initially speak the right password to gain access to a system. The system not only verifies if the correct password has been said but also focused on the authenticity of the speaker. The ultimate goal is do have a system which does a Speech, Iris, Fingerprint Recognition to implement access control.

Developing an Isolated Word Recognition System in MATLAB

PRODUCTS USED:

- *MATLAB (MATHWORKS)*
- *DATA ACQUISITION TOOLBOX*
- *SIGNAL PROCESSING TOOLBOX*
- *STATISTICS TOOLBOX*

The development workflow consists of three steps:

- *SPEECH ACQUISITION*
- *SPEECH ANALYSIS*
- *USER INTERFACE DEVELOPMENT*

Acquiring Speech

For training, speech is acquired from a microphone and brought into the development environment for offline analysis. For testing, speech is continuously streamed into the environment for online processing. During the training stage, it is necessary to record repeated utterances of each digit in the dictionary. For example, we repeat the word 'one' many times with a pause between each utterance.

FOLLOWING MATLAB CODE CAPTURES 10 SECONDS OF SPEECH WITH 8000 SAMPLES PER SECOND:

```
Fs = 8000;                    % Sampling Freq (Hz)

 Duration = 10;               % Duration (sec)

 y = wavrecord(Duration*Fs,Fs);
```

We save the data to disk as 'mywavefile.wav':

MATLAB CODE FOR CAPTURING SPEECH DATA

```
% Define system parameters

framesize = 80;                    % Framesize (samples)

Fs = 8000;                        % Sampling Frequency (Hz)

RUNNING = 1;                      % A flag to continue data capture

% Setup data acquisition from sound card

ai = analoginput('winsound');

addchannel(ai, 1);

% Configure the analog input object.

set(ai, 'SampleRate', Fs);

set(ai, 'SamplesPerTrigger', framesize);

set(ai, 'TriggerRepeat',inf);

set(ai, 'TriggerType', 'immediate');

% Start acquisition

start(ai)

% Keep acquiring data while “RUNNING” ~= 0

while RUNNING

 % Acquire new input samples

 newdata = getdata(ai,ai.SamplesPerTrigger);

 % Do some processing on newdata

…

<DO _ SOMETHING>

…
```

```
 % Set RUNNING to zero if we are done
 if <WE _ ARE _ DONE>
 RUNNING = 0;
 end
end
% Stop acquisition
stop(ai);
% Disconnect/Cleanup
delete(ai);
clear ai;
```

MATLAB CODE FOR READING A SPEECH SAMPLE FRAME BY FRAME

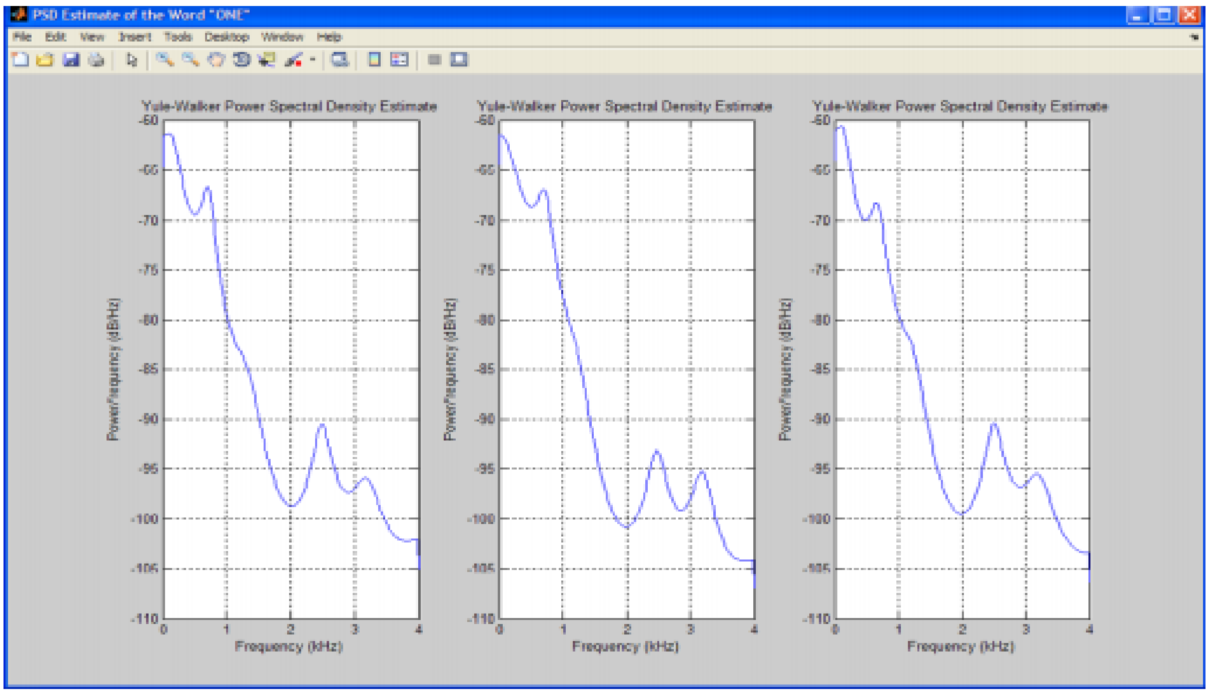
```
% Define system parameters
seglength = 160;                       % Length of frames
overlap = seglength/2;                 % # of samples to overlap
stepsize = seglength - overlap;        % Frame step size
nframes = length(speech)/stepsize-1;
% Initialize Variables
samp1 = 1; samp2 = seglength;           %Initialize frame start and end
for i = 1:nframes
 % Get current frame for analysis
 frame = speech(samp1:samp2);
 % Do some analysis
…
<DO _ SOMETHING>
…
 % Step up to next frame of speech
 samp1 = samp1 + stepsize;
 samp2 = samp2 + stepsize;


end
```

Developing the Acoustic Model

A good acoustic model should be derived from speech characteristics that will enable the system to distinguish between the different words in the dictionary.
We know that different sounds are produced by varying the shape of the human vocal tract and that these different sounds can have different frequencies. To investigate these frequency characteristics we examine the power spectral density (PSD) estimates of various spoken digits. Since the human vocal tract can be modeled as an all-pole filter, we use the **Yule-Walker** parametric spectral estimation technique from Signal Processing Toolbox™ to calculate these PSDs.
After importing an utterance of a single digit into the variable 'speech', we use the following MATLAB code to visualize the PSD estimate:

```
order = 12;
nfft = 512;
Fs = 8000;
pyulear(speech,order,nfft,Fs);
```



Since the Yule-Walker algorithm fits an autoregressive linear prediction filter model to the signal, we must specify an order of this filter. We select an arbitrary value of 12, which is typical in speech applications.
We can see that the peaks in the PSD remain consistent for a particular digit but differ between digits. This means that we can derive the acoustic models in our system from spectral features.
From the linear predictive filter coefficients, we can obtain several feature vectors using Signal Processing Toolbox functions, including reflection coefficients, log area ratio parameters, and line spectral frequencies.
One set of spectral features commonly used in speech applications because of its robustness is Mel Frequency Cepstral Coefficients (MFCCs).