THE FLORIDA STATE UNIVERSITY

COLLEGE OF ENGINEERING

SPEAKER INDENTIFICATION BASED ON AN INTEGRATED SYSTEM COMBINING CEPSTRAL FEATURE EXTRACTION AND VECTOR QUANTIZATION

By

JOSE BORIS SANCHEZ

A Thesis submitted to the
Department of Electrical Engineering
in partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Spring Semester, 2005

The members of the committee approve the thesis of Jose Boris Sanchez defended on 04/07/2005.

Anke Meyer-Baese
Professor Directing Thesis

Leonard Tung
Committee Member

Simon Foo
Committee Member

Leonard Tung, Chair, Department of Electrical and Computer Engineering

Ching-Jen Chen, Dean, College of Engineering

The Office of Graduate Studies has verified and approved the above named committee members.

Dedicated to my parents for their everlasting support, Dr. Walker for offering me opportunities, Dr. Meyer-Baese for her priceless help, and God for making it all possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The purpose of the research conducted in the thesis is to study the effectiveness of a text-dependent identification system making use of cepstral coefficients and vector quantization. The identification system will make use of Mel-frequency cepstral coefficients (MFCC) and the effects of utilizing these vs. just cepstral coefficients will be examined. MFCC speech features are to be extracted from voice recordings and subjected to vector quantization. The data resulting from the analysis will serve as the key characteristic in identifying the person to whom the recorded voice belongs.

INTRODUCTION

Speaker identification is one of the two categories of speaker recognition, with speaker verification being the other one. The main difference between the two categories will now be explained. Speaker verification performs a binary decision consisting of determining whether the person speaking is in fact the person he/she claims to be or in other words verifying their identity. Speaker identification performs multiple decisions and consists comparing the voice of the person speaking to a database of reference templates in an attempt to identify the speaker. Speaker identification will be the focus of the research in this case.

Speaker identification further divides into two subcategories, which are text-dependent and text-independent speaker identification. Text-dependent speaker identification differs from text-independent because in the aforementioned the identification is performed on a voiced instance of a specific word, whereas in the latter the speaker can say anything. The research will consider only the text-dependent speaker identification category.

System Overview

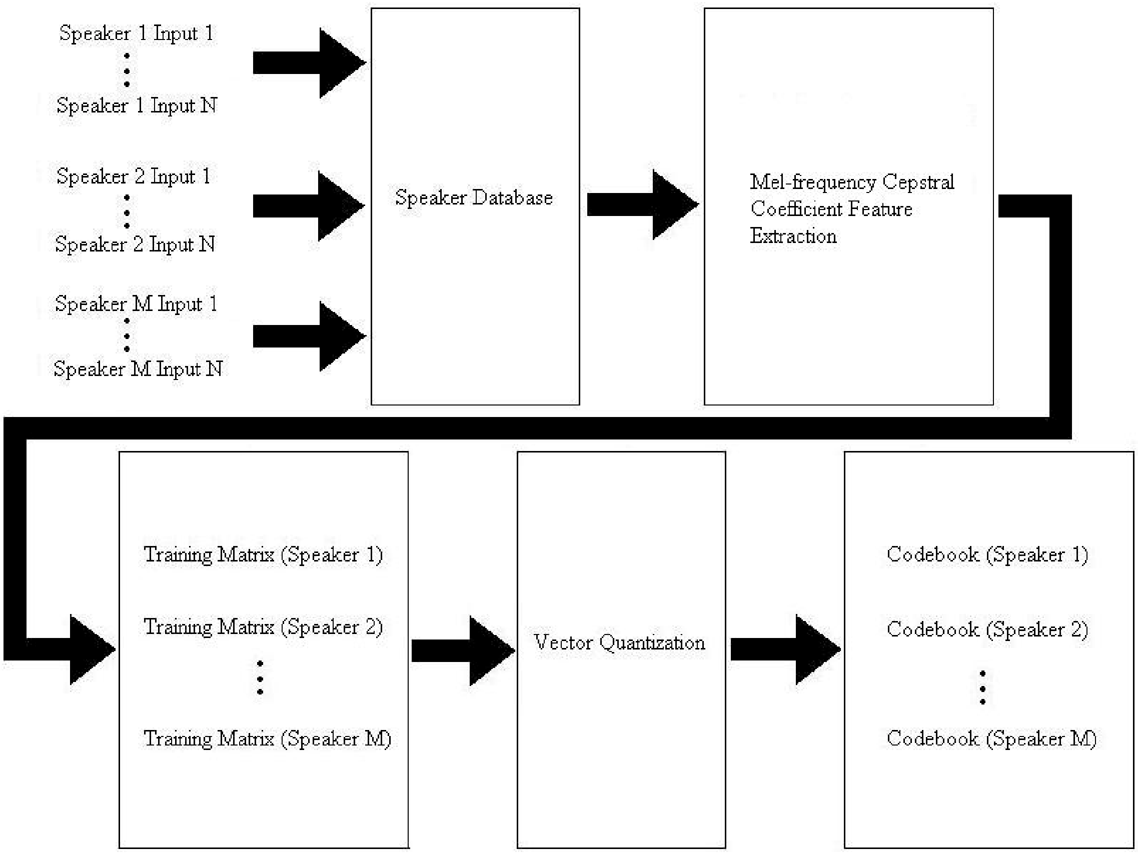

Figure 1. Codebook Creation

The figure above is a flow chart representation of the general steps used in the formation of codebooks to represent the speakers.

The first step is to create a speaker database containing digitized speech recordings of all the people that are to be identified. A database composed of 5 individuals of distinct sexes was created from recordings of instances in which the word "Boris" was spoken. These recordings then underwent Mel-frequency cepstral coefficient feature extraction. Training matrices for each of the speakers were later formed from available MFCC matrices obtained in the previous step. The training matrices were then utilized to obtain codebooks that would serve as references for each speaker after applying vector quantization.

The front-end of the system was mentioned, so the identification portion follows. After a digitized representation of an instance of "Boris" being spoken is obtained, the next stage of the process is to identify the speaker. The audio input has to be processed and the Mel-frequency cepstral coefficients need to be obtained. This MFCC matrix is then to be matched to all the available speaker codebooks that have been stored. The codebook that returns the lowest quantization error should belong to the speaker whose voice is contained in the audio input file.

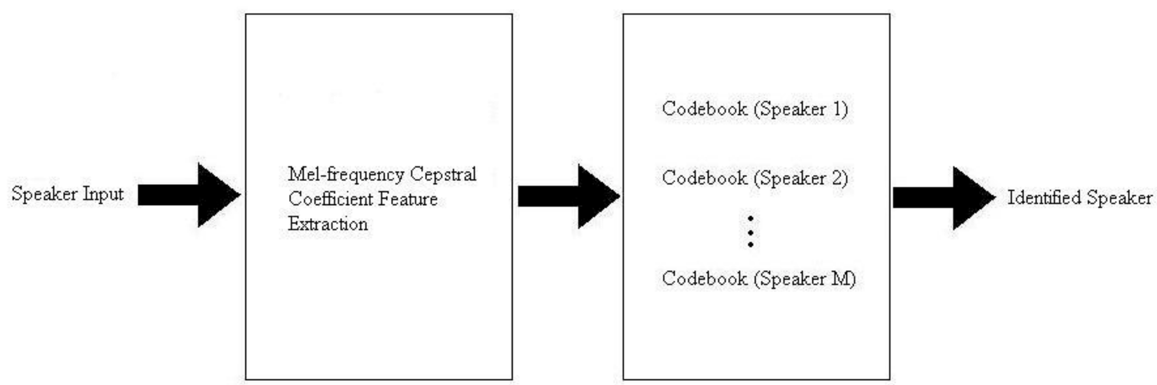

Figure 2. Speaker Identification

The figure above shows the stages involved in the speaker identification process.

The following subsections detail the signal processing involved for extracting the Mel-frequency cepstral coefficients:

Framing

Speech is a nonstationary signal, but when segmented into parts ranging from 10-40 msec, these divisions are quasi-stationary. For this reason the speech input is to be divided into frames before feature extraction takes place. The selected properties for the speech signals are a sampling frequency of 16 kHz, 8-bit monophonic PCM format in WAV audio. The chosen frame size is of 256 samples, resulting in each frame containing 16 msec portions of the audio signal. If the final frame of the audio signal is less than 256 samples long, the frame is zero padded to allow processing consistency and little added expense. Each of the frames is then normalized.

Windowing

All the frames are then multiplied by a window function. Each of the frames is separated by 128 samples or in other words overlaps by 50%, so that each sample is included in two frames for processing. It is the nature of a typical window function, such as a Hamming or Hanning window, that serves as the reason for this overlap. The use of the window function reduces the frequency resolution by 40%, so the frames must overlap to permit tracing and continuity of the signal. The motive for utilizing the windowing function is to smooth the edges of each frame to reduce discontinuities or abrupt changes at the endpoints. The windowing serves a second purpose and that is the reduction of the spectral distortion that arises from the windowing itself. A Hamming window, characterized by

$$W_H(n) = 0.54 - 0.46 \cos (2n\pi / N - 1)$$

was used for this process and can be seen in Figure 3. The nominal frequency resolution of the calculate spectrum is $\Delta f = 1 / TN = 1 / (1 / 16000)(256) = 62.5\ Hz$

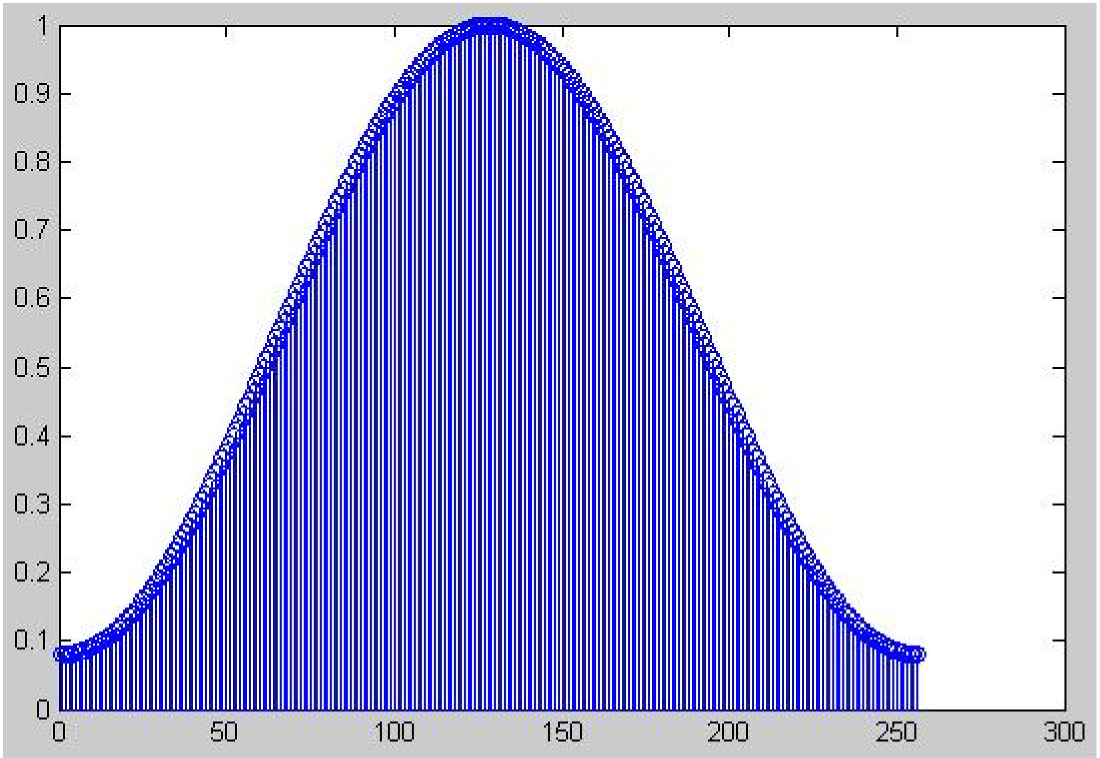

Figure 3. Hamming window

The Hamming window applied on audio frames may be seen above. The borders of the window serve to smooth out the edges of the frames.

Fast-Fourier Transform

The frame size is not a fixed quantity and therefore can vary depending on the resulting time portion of the audio signal. The reason that the number of samples was selected as 256 is that it is a power of 2, which enables the use of the Fast-Fourier Transform. The FFT is a powerful tool since it calculates the DFT of an input in a computationally efficient manner, saving processing power and reducing computation time. The FFT is characterized by the following

$$X(k) = \Sigma\, x(j)\, w_N^{(j-1)(k-1)}$$

, where x(j) is the j^{th} sample, $w_N = e^{(-2\pi i)/N}$. The operation results in the spectral coefficients of the windowed frames.

Mel-scale Filterbank Frequency Transformation

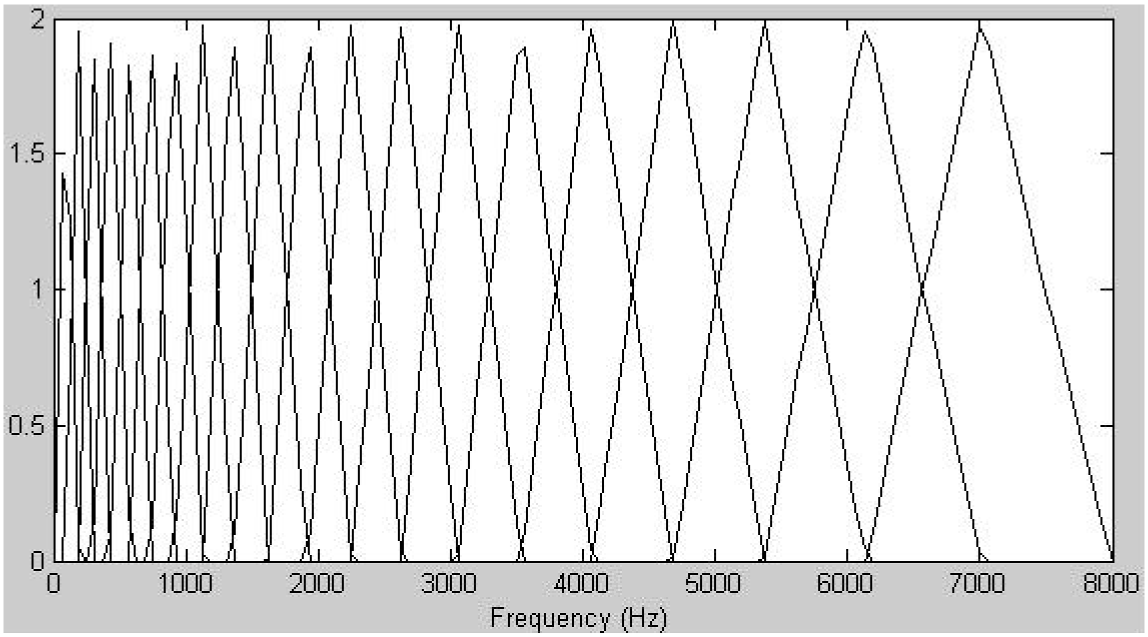

Figure 4. Mel-scale Filterbank

The figure above depicts the Mel-scale Filterbank applied on the processed frames. It is composed of 20 triangular filters equally spaced on logarithmic scale.

Mel-cepstral coefficients are the features that will be extracted from speech during this research. The key difference between MFCCs and cepstral coefficients lies in the processing involved when extracting each of these characteristics of a speech signal. The process of obtaining Mel-cepstral coefficients involves the use of a Mel-scale filter bank. The purpose of such a filter bank will be explained in a later section of the chapter.

The spectral coefficients of each frame are then converted to Mel scale after applying a filterbank. The Mel-scale is a logarithmic scale resembling the way that the human ear perceives sound. The filterbank is composed of triangular filters that are equally spaced on a logarithmic scale, as may be viewed on Figure 4.

The Mel-scale is represented by the following

$$Mel(f) = 2595 \log_{10} (1 + f / 700)$$

, where f is frequency. The spectral coefficients of the frames are binned or multiplied by the filter gain and accumulating the results. This has as an outcome each bin containing the spectral magnitude in the filterbank channel. 20 filters were used to create the filterbank and its use renders Mel-spectral coefficients.

Discrete Cosine Transform

The Discrete Cosine Transform is applied to the log of the Mel-spectral coefficients to obtain the Mel-Frequency Cepstral Coefficients. The Discrete Cosine Transform is described defined by the following

$$Y(k) = w(k)\ \Sigma x(n) \cos(\pi(2n - 1)(k - 1) / 2N), \text{ where } k = 1, 2, \ldots, N,$$

$$x(n) \text{ is the } n^{th} \text{ sample, and } w(k) = 1 / sqrt(N) \ \text{ for } k = 1$$

$$= sqrt(2 / N) \ \text{ for } 2 \leq k \leq N$$

Only the first 12 coefficients of each frame are kept, since most of the relevant information is kept amongst those at the beginning. The first 12 coefficients (1^{st} frame) can be discarded since they are the mean of the signal and hold little information. The use of the DCT minimizes the distortion in the frequency domain and is efficient in its calculation since an N-point DCT can be carried out using a symmetric 2N-point FFT.

Cepstral Mean Subtraction

This process is incorporated to reduce the effects of additive noise, such as those incurred from a microphone, environment, transmission lines, etc. The technique takes advantage of the fact that the multiplicative effects become additive in the log cepstral domain. It is simple yet effective, solely subtracting the long-term cepstral mean from the cepstral coefficients to help remove the undesired forces.

Process Visualization

In this section we will visualize the results obtained from some of the main parts of the feature extraction process, having recently viewed the details of the procedure. Figure 5 will serve as the audio signal intended for the analysis in this case.

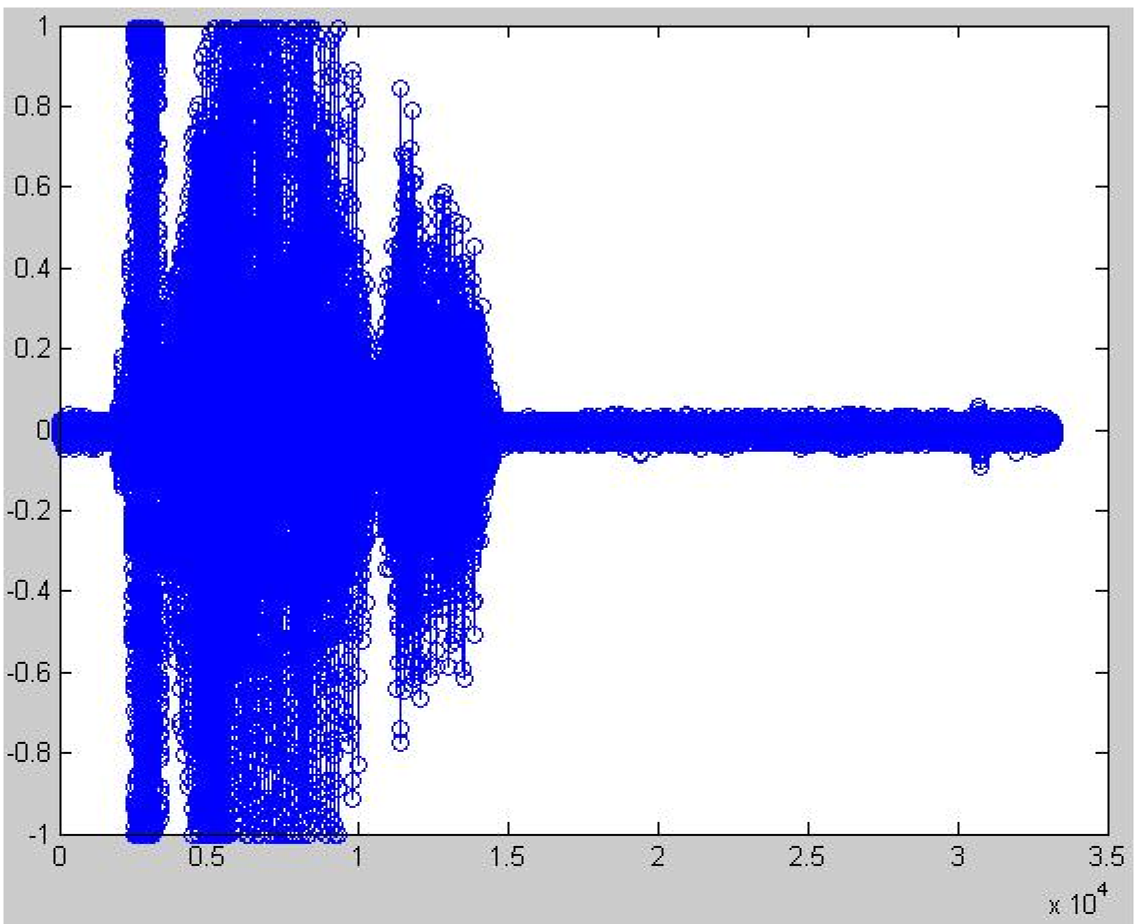

Figure 5. Digital Audio Signal

This audio sample represents the spoken instance of the name "Boris". It will serve as the input of the feature extraction in order to visualize the results of the processing involved.

The next step is to frame the audio sample into portions of a predetermined size. This is done to process the frames taking advantage of the quasi-stationary property of the frames, given a properly selected frame size.

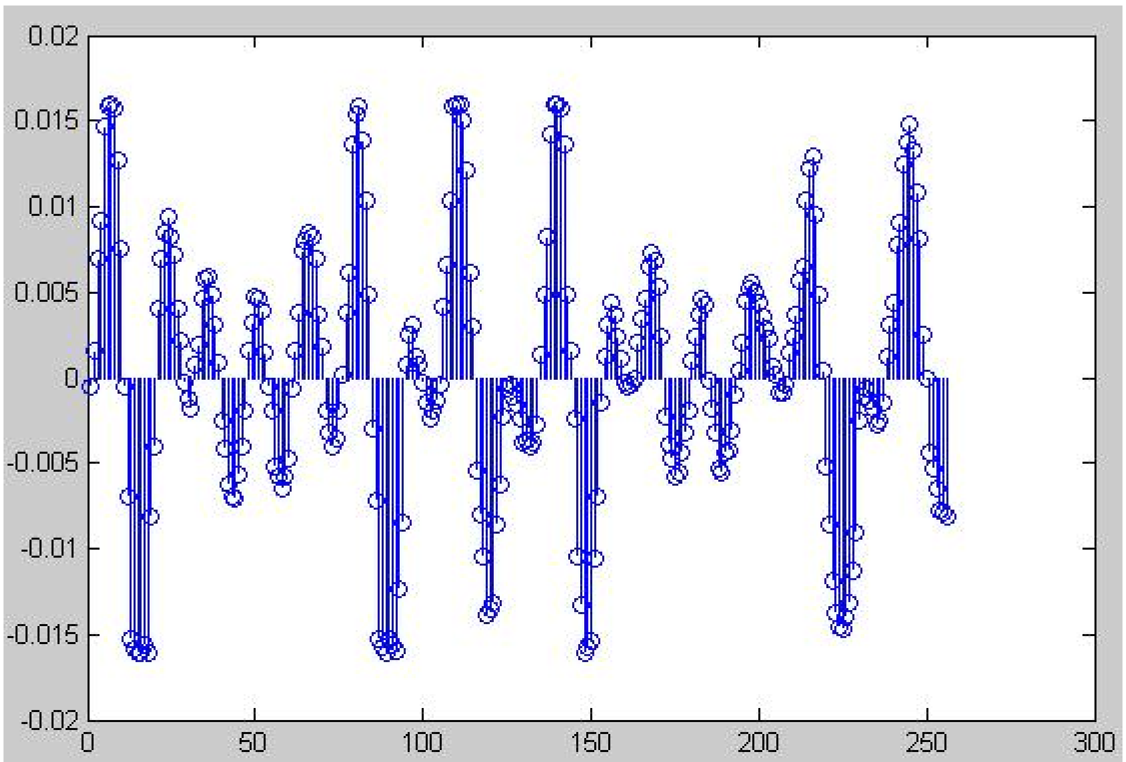

Figure 6. Frame

Shown above is a frame belonging to the digital audio signal previously shown. The frame size is composed of 256 samples for an equivalent of 16msec of the audio signal according to the properties that were selected.

Next a windowing function is applied to the frames. In this case a Hamming window was used on the individual frames to smooth out the frame edges and reduce spectral distortion. Figure 7 shows the result of applying the Hamming window to the frame size shown above. The Hamming window itself can be seen in Figure 3.

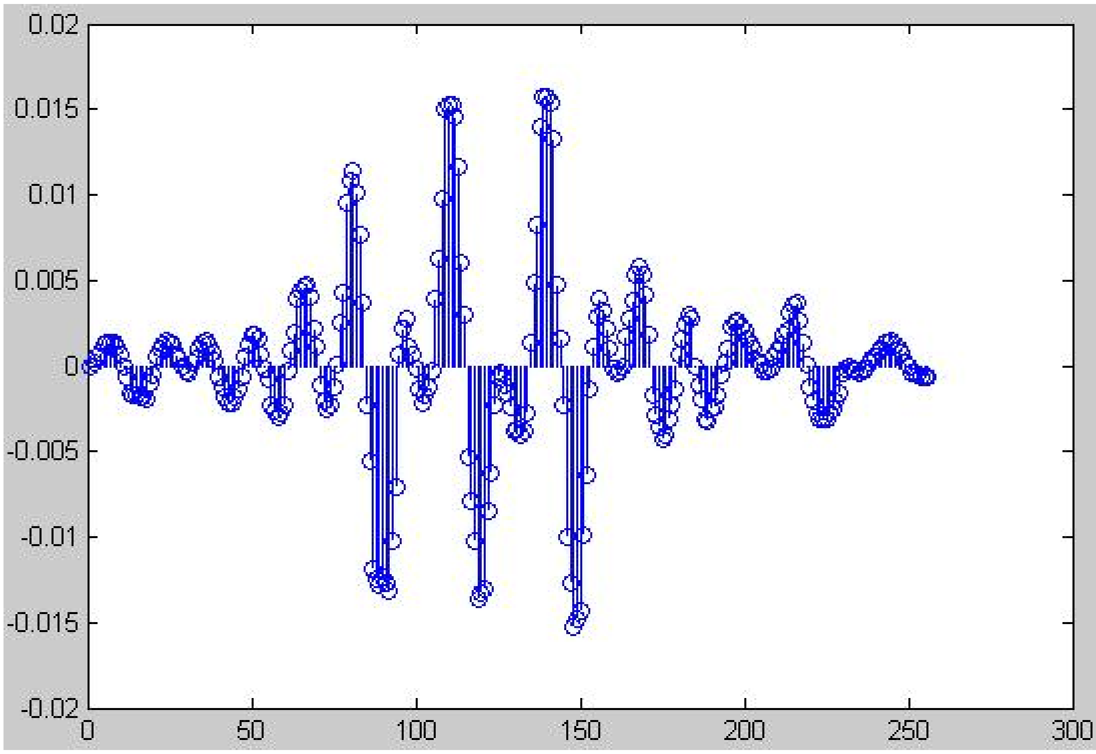

Figure 7. Windowed Data

The Hamming window was applied to the selected frame, resulting in the data shown above.

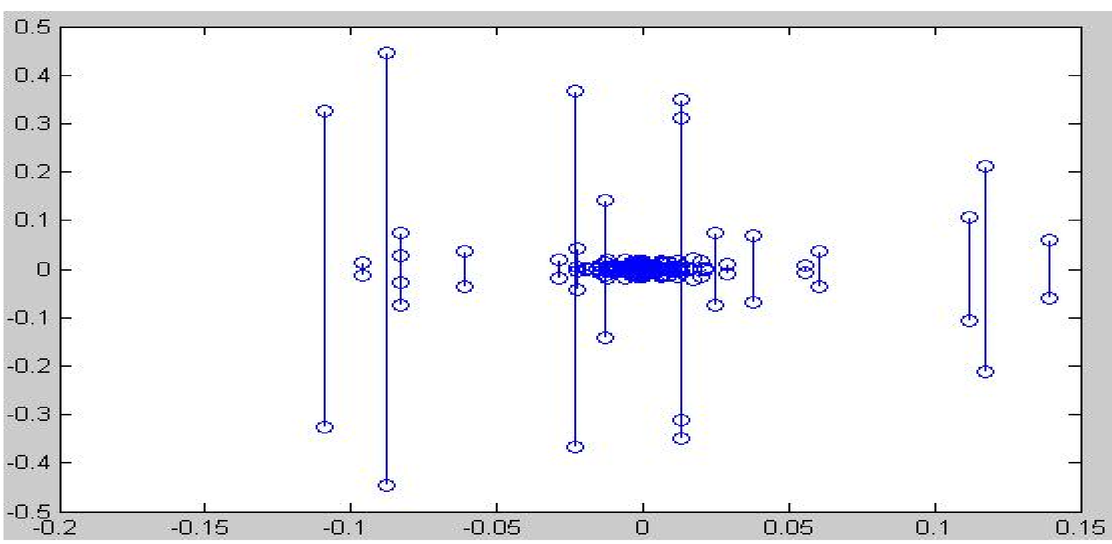

Figure 8. Spectral Coefficients

The spectral coefficients of the windowed frame are displayed above. These were computed through the use of the Fast-Fourier Transform.

The windowed data is then to undergo the Fast Fourier Transform in order to compute the spectral coefficients of the windowed frame. This was one of the advantages of the selected frame size, which was chosen as a power of 2 for the purpose of utilizing this tool. The spectral coefficients resulting from the windowed frame can be seen in Figure 8.

Then the spectral coefficients are processed with a Mel-scale filterbank to convert these to the Mel scale. The filterbank used may be viewed in Figure 4.

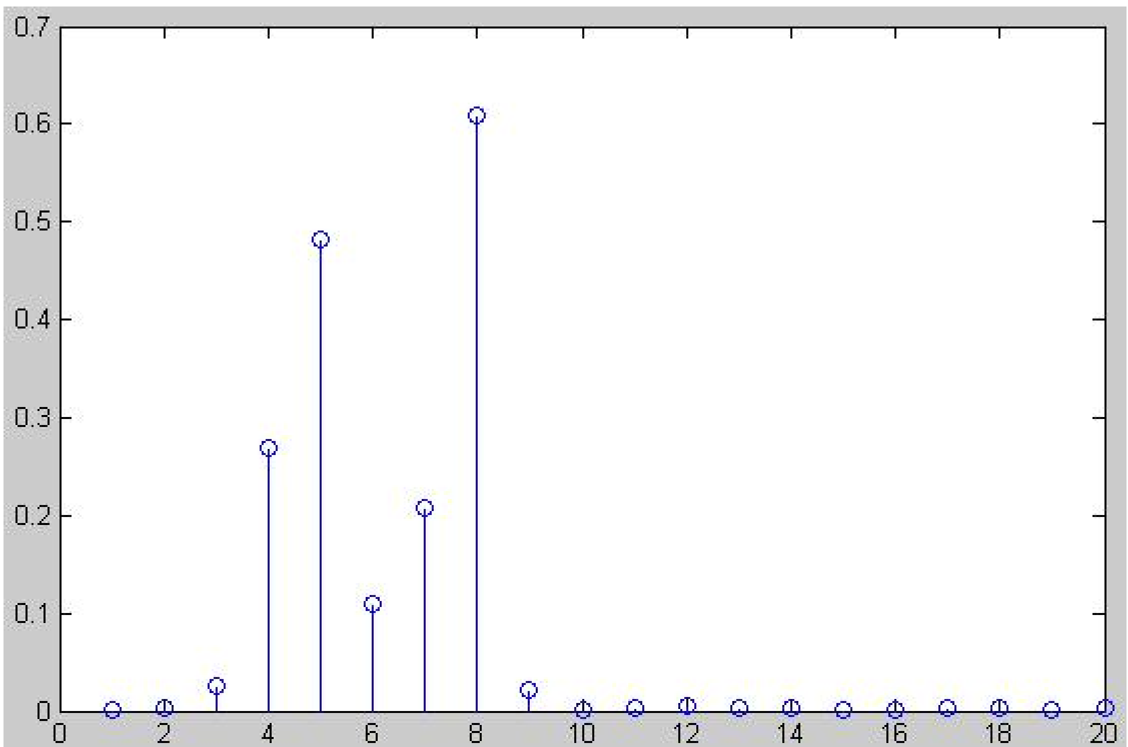

Figure 9. Mel Spectral Coefficients

These are the Mel spectral coefficients resulting from applying the Mel-scale filterbank to the spectral coefficients of the windowed frame.

The logarithms of these Mel spectral coefficients are then transformed to the frequency domain with the Discrete Cosine Transform. Of each frame only the first 12 coefficients are kept to avoid extra data containing less important information.

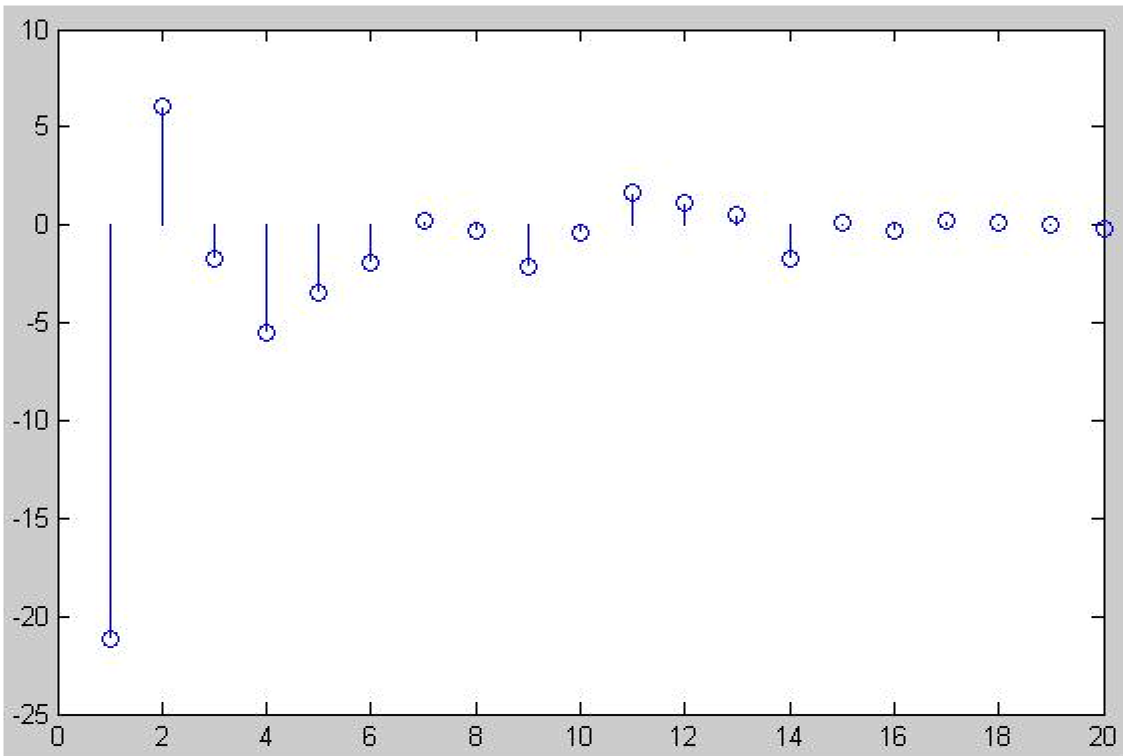

Figure 10. Mel-frequency cepstral coefficients

These are the Mel-frequency cepstral coefficients computed from the selected frame.

The flow chart of the feature extraction process is portrayed in Figure 11. The Mel-frequency cepstral coefficients of the whole audio sample can be seen in Figure 12.

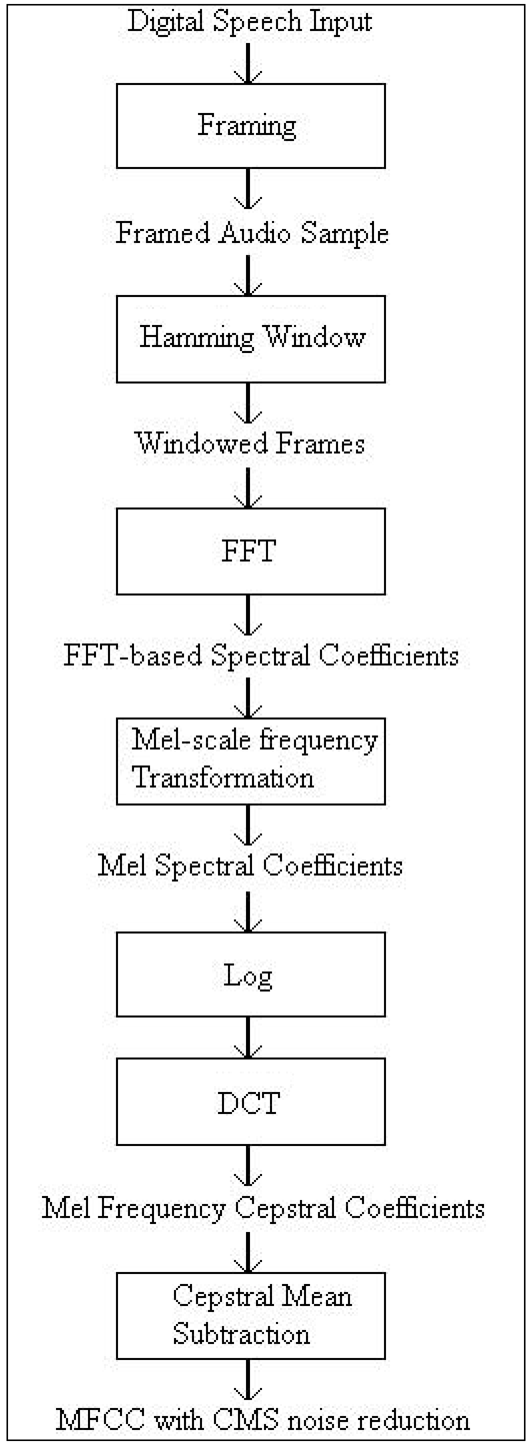

Figure 11. MFCC Feature Extraction

The diagram above is a flow chart listing the procedure followed in order to extract the Mel-frequency cepstral coefficients from a digitized audio signal.

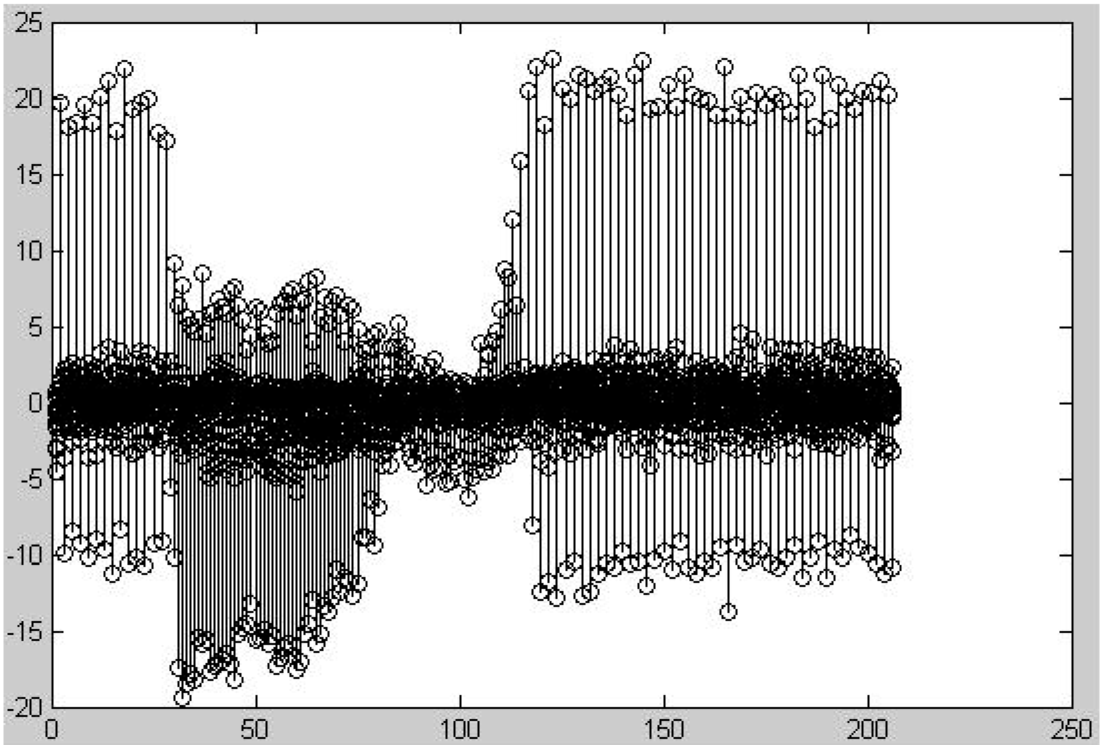

Figure 12. MFCC of Audio Signal

Shown above are the Mel-frequency cepstral coefficients of the entire audio signal. Only the first 12 of each frame were kept, since they contain the more valuable information.

CLASSIFICATION OF SPEAKERS BASED ON VECTOR QUANTIZATION

Vector quantization gains its name from the fact that it is a quantization method that deals with vectors rather than individual samples or scalars. A training pattern is formed by concatenating the MFCCs extracted from the available training samples. Depending on the size determined for the codebook, training patterns are chosen to form the code vectors that make up a codebook. One detail to point out is that both codebook and the training pattern are matrices. The use of the term *vector* in the context of the vector quantization used in the research is equivalent to a row of a matrix. The codebook can be generated by either randomly selecting the code vectors from the training data or clustering training data and calculating centroids that will create the codebook (Lloyd's algorithm).

Lloyd's Algorithm

Lloyd's algorithm was chosen as the method with which to carry out the vector quantization for this research. Following is the description of this algorithm and details regarding its utilization in this project:

Initialization

Each of the training samples underwent MFCC calculation and only the first 12 coefficients of each processed frame were kept. The MFCCs for a particular speaker were stored consecutively as rows of a training matrix (size T * 12), which serves as the input to give rise to a codebook representing that speaker. Random rows of the training matrix were selected to form an initial codebook.

Vector Coding

During this phase of the algorithm, each vector in the training matrix is categorized with respect to the codebook. The manner in which this is carried out is by calculating the Euclidean distance between a given training vector and each of the code vectors in the codebook. Once the code vector that minimizes this criterion is identified, the training vector is labeled with the entry or position of that code vector. This is done for all the training vectors available for a specific speaker.

Codebook Updating

Once all the training vectors have been labeled with an index linking them to the proper code vectors, the codebook is updated. All the training vectors that have been labeled with the index of a particular code vector represent a cluster. This means that if there are M code vectors in the codebook, there will be M clusters in the training matrix. The centroids for all given clusters are calculated and each centroid replaces the code vector positioned at the index indicated by vectors in a cluster.

Quantization Error Calculation

This part of the process requires the total distortion, deviation from training material, of the codebook to be computed. The way this is done is by calculating the Euclidean distance between each of the training vectors and each of the code vectors and adding these distances together. The summation represents the total quantization error of the codebook. Lloyd’s algorithm is an iterative process and the total quantization error is the factor that determines the times that the algorithm will be repeated. Given that the algorithm runs at least two times, the quantization error of the previous time is compared to the newly computed quantization error. Only if the present quantization error is less than the previous one will the speaker’s codebook be modified. The algorithm will be repeated starting form the second process, vector coding. The initialization will only take place the first time that the algorithm runs. In the case that the previous quantization error is less than the present one, the algorithm will terminate execution.

Considerations

An important issue to consider when dealing with Lloyd’s algorithm is that the distortion sometimes converges to a local minimum, which may be significantly worse than the global minimum. More specifically the distortion tends to move towards the local minimum closest to the initial codebook. For this reason the algorithm can be carried out several times with different initial codebooks. Then the quantization errors resulting from all of these may be compared to each other in an effort to select the codebook that renders the lowest quantization error. This would constitute the codebook that serves as a reference for a particular speaker. Vector quantization is deemed as an "efficient coding method because it utilizes the statistical occurrence or the probability distribution of the source, no matter how varied it is".[1]

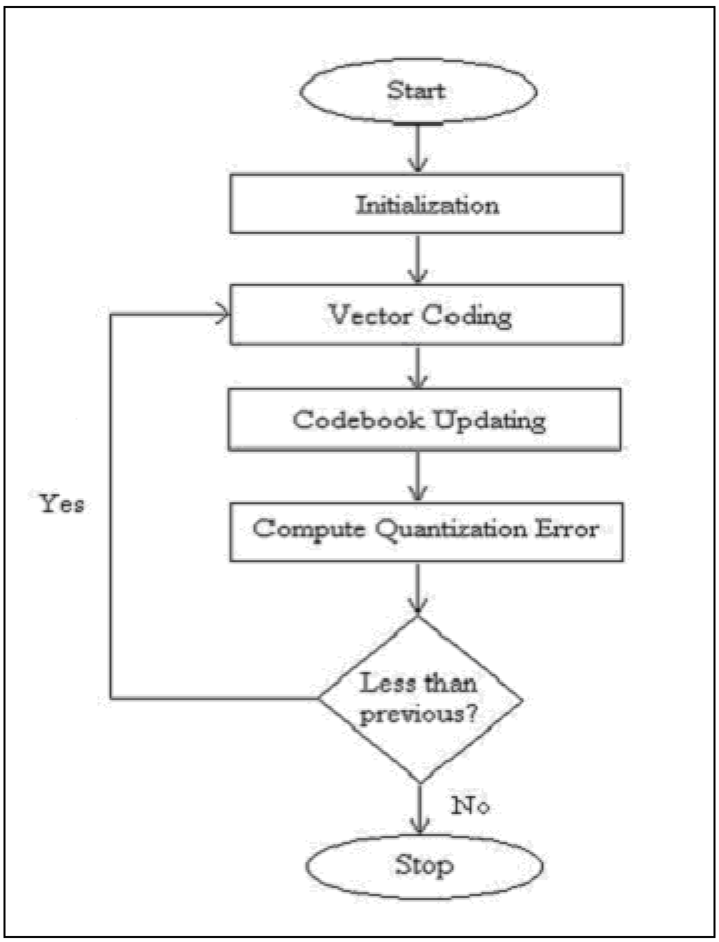

Figure 13. Vector Quantization Flow Chart

Shown above is a flow chart detailing the steps involved in the iterative process of vector quantization.

---

[1] Furui, Sadaoki. Digital Speech Processing, Synthesis, and Recognition (New York: Marcel Dekker, 2001) 177.

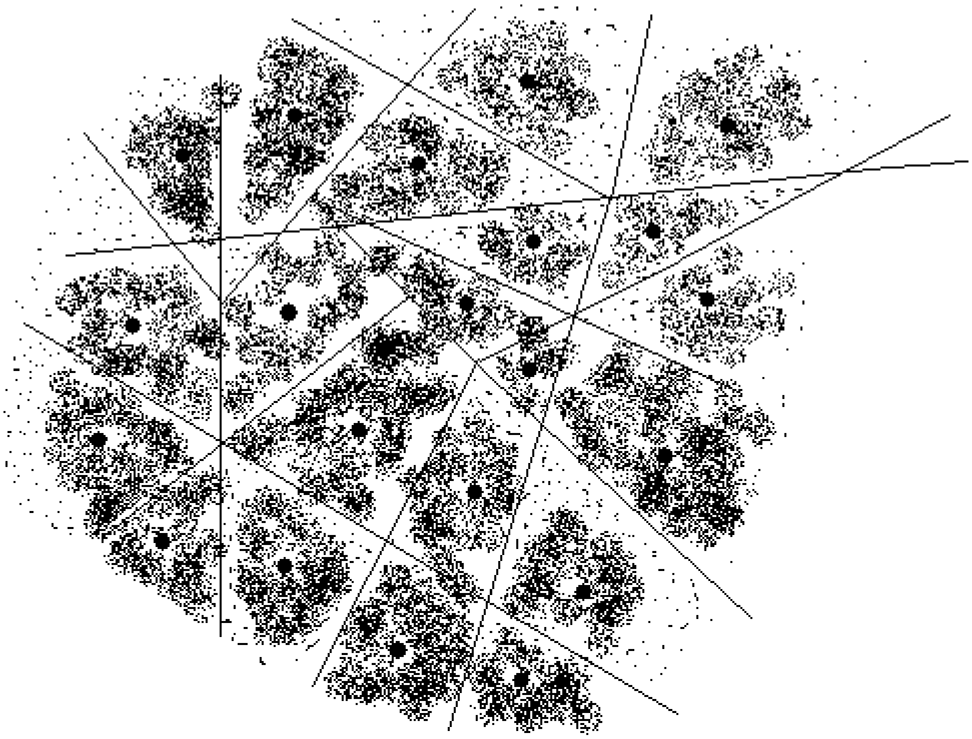

Figure 14. Vector Quantization

Shown above is an illustration of vector quantization.

Figure 14 depicts what the result of vector quantization having concluded might look like. The whole dark spots represent the code vectors of the codebook and the surrounding points symbolize the training vectors. As mentioned earlier in Lloyd's algorithm, these centroids will be recalculated for every iteration of the algorithm in order to produce a better representation of the training matrix. This process will be carried out for each speaker requiring a codebook

RESULTS

Several test cases have been examined and analyzed in order to evaluate the performance of the system under the selected specifications. The details of each of the cases and the results obtained from each instance will follow shortly. Tabulated information obtained from simulations is provided to illustrate the findings and to support the observations that were obtained from analysis.

Bits per Audio Sample: 8-bit vs. 16-bit

The bit rate of the samples was not a fixed parameter when the research on mention began. 8-bit and 16-bit bit rates, being standard and commonly used bit rates for WAV audio, were both considered for this study. The use of 16-bit audio rate shows no clear advantage over the use of the 8-bit rate. This observation allows for the compromise of utilizing the 8-bit rate and to reap the benefits of doing so. The clear gain from using the smaller rate would be the size of an audio file being reduced in half. Tabulated data may be seen documented on Table 1.

Gains deriving from the file size reduction concretize both in performance and specifications of the system. The lower bit rate reduces not only the storage required for each of the files in the database but also the cost of dollar per byte for a predetermined storage system. Less processing requirements also arise from the use of the lower bit rate, since now operations would be performed on less data. Lower storage and processing requirements better allow for portability and is limited only by the amount of users and the purpose of the system.

Codebook Size

The size of the codebook is definitely a variable parameter and an intrinsic element of the system bearing special significance on performance. This characteristic determines the fluctuation of the quality of the codebook that can be obtained. The larger the number of code vectors in the codebook, the better the quality of codebook that can be obtained. The quality of a codebook, as explained in the vector quantization chapter, is measured in the amount of quantization error obtained from the codebook.

Increasing the size of the codebook also reduces the possible fluctuation from obtaining different codebooks. This means that for a fixed codebook size, if different codebooks are obtained, the difference in quantization error amongst the computed codebooks is decreased. In other words, one may observe that the quantization errors of codebooks obtained for a particular speaker tend to converge when increasing the codebook size. Therefore, not only is it possible to obtain a codebook with lower quantization error by increasing the allowable size, but it also facilitates the selection of a codebook to reference a given speaker. Information supporting these findings can be found in the contents of Table 1.

Of course a limit does exist on both the lowest quantization error that a codebook can produce and how much the quantization errors of codebooks of a specific size will converge. Once again, compromise is the key when selecting the codebook size. It should be selected in such a way as to reduce the variability of the codebooks to an acceptable degree. Not only would this ensure that an approximately equal quality codebook would be obtained if the algorithm is run again, but it would be implied that a good quality codebook is obtained.

Table 1 is organized in such a way as to view the effects caused by selecting either 8-bit or 16-bit samples and by varying the codebook size. Quanterror represents the quantization error of one of seven random codebooks that were computed given the bit rate and codebook size being considered. Melc11 – 15 represent the selected audio samples to be analyzed and the entries under these are the quantization errors calculated under the codebook being considered. The codebooks are arranged in ascending order of quantization error and the greatest and smallest value found under each bit rate – codebook size section represent the worst and best codebook, respectively, found during the entire trial run for each individual setting.

Table 1. Bit Rate and Codebook Size Variation

Bit Rate	Codebook Size	quanterror	melc11	melc12	melc13	melc14	melc15
		7403.4	662.3135	670.8778	630.6184	741.6802	625.2862
		7442.4	665.9378	665.0285	620.6375	752.9892	621.3266
		7479.3	676.311	668.2601	627.0081	745.369	622.4185
8-bit	16	7543.6	685.6532	690.5936	621.1015	730.176	626.0389
		7727.1	724.1279	700.9927	642.9583	758.027	640.5575
		8708.2	757.6836	809.2475	722.7196	809.3416	703.401
		8962.2	731.2204	828.0621	770.8614	861.2981	734.7797
		7587.9	673.4865	691.9711	651.7115	771.099	642.6008
		7625.5	681.9483	683.3696	648.563	782.5508	640.5004
		7722.8	689.2754	703.5831	643.2453	765.3998	650.3183
16-bit	16	7754.4	698.2855	692.7911	642.1754	773.1164	638.45
		7828.2	726.0082	701.6089	657.3125	785.0919	655.0003
		8801.8	753.3505	836.4536	786.1685	862.4509	749.5415
		8844.2	768.1436	825.4591	737.9057	832.4586	723.7415
		6599.3	614.4787	601.4302	566.0075	663.3809	565.0855
		6609.1	592.0718	607.4443	563.0291	669.7905	565.5186
		6721.3	610.1742	614.0781	564.9272	673.3739	584.7697
8-bit	32	6769.9	629.833	618.1962	572.1185	667.6273	575.2208
		6776.7	629.033	609.4709	564.7077	668.9941	561.6368
		6967.2	644.8216	642.5816	586.6114	687.7294	590.5472
		6990.6	644.1612	633.3879	587.0862	691.0203	587.7011

Table 1 – continued

Bit Rate	Codebook Size	quanterror	melc11	melc12	melc13	melc14	melc15
		6853.8	617.5452	631.5733	588.9484	694.7809	595.5363
		6955.2	631.3569	637.5351	587.3764	708.2142	590.9948
		7047.4	649.9246	641.5694	605.3191	714.5365	606.5954
16-bit	32	7103.3	648.6034	644.6184	604.0324	715.7956	603.621
		7139.6	661.2415	653.3208	597.1991	691.4475	598.3642
		7238.7	681.3063	666.2509	622.8955	719.708	609.1196
		7286.4	688.1468	685.1471	605.4294	705.894	603.6658
		6051.1	573.7114	571.2691	512.6252	626.9086	521.7904
		6073.9	569.9165	568.4984	519.8085	625.3672	528.1903
		6104.5	569.0073	572.6829	529.8302	638.9574	536.9327
8-bit	64	6126.1	560.9315	578.7092	519.1503	613.8749	523.8306
		6128	578.6719	565.1015	526.9341	615.3195	530.1313
		6166.2	568.4588	581.4421	524.8342	624.5748	532.2025
		6199.2	577.3067	570.254	532.0775	619.3932	529.5695
		6230.1	573.4783	581.5227	539.6292	645.4034	538.026
		6236.4	575.921	592.1379	537.0121	646.8932	540.364
		6244	575.3414	572.2399	550.8439	654.292	546.6496
16-bit	64	6257.6	579.4791	583.9112	542.6607	630.1071	551.5456
		6291.1	586.2551	583.542	543.5147	644.6255	544.9136
		6300.6	583.3304	589.0326	548.152	653.0267	551.0065
		6303.2	604.4416	587.9475	543.1069	652.2705	539.4006

The effects of varying both bit rate and codebook size for audio samples of a given speaker may be seen in the data recorded in the table above.

Cepstral vs. MFCC

An interesting comparison to make is that of the efficiency of speaker identification resulting from the use of cepstral coefficients or Mel-frequency cepstral coefficients. Codebooks were formed for both cepstral coefficients and MFCCs obtained from the same set of audio samples in order to make the comparison between the types of coefficients. The results that were obtained indicate that the Mel-frequency cepstral coefficients prove to be the better choice of the two coefficient types.

The outcome of the analysis may be found in the data listed in Table 2. The table is divided into the type of coefficient used and each quantization error listed represents a codebook generated using the coefficient type it is listed to the right of. The entries for cc# represent the cepstral coefficients extracted from the file represented by #. The same reasoning applies to the melc# entries with the difference that MFCCs were extracted in that case.

The comparison of the performance of cepstral coefficients vs. Mel-frequency cepstral coefficients was done with the bit rate of 8 bits per sample and a codebook size of 64 due to the advantages discussed previously. The same set of audio files were used in the creation of the codebooks for both types of cepstral coefficients. Codebooks represented by the quantization errors listed were randomly selected, except for the codebooks resulting in the lowest and highest errors. These are the minimum and maximum values found during the respective trial runs for each coefficient type. All other codebooks computed from the trial runs, but not listed in the tables fell under the range specified by these boundaries.

Table 2. Cepstral Coefficients vs. MFCC

Coefficient Type	Quantization Error	cc2	cc7	cc13	cc21	cc23
Cepstral	12445	1308.1	1350.3	1149	1214.7	1148.4
	12482	1310	1343.3	1141.2	1217.3	1150.5
	12586	1278.6	1342.8	1154.1	1248	1188.4
	12605	1342	1361	1176.4	1212	1125.7
	12759	1340.9	1350.8	1186.2	1241.6	1130.5

Table 2 – continued

Coefficient Type	Quantization Error	melc2	melc7	melc13	melc21	melc23
MFCC	5943.2	645.2373	654.6652	548.0658	597.9409	538.7766
	5983.4	656.1945	637.3722	566.6013	609.6099	545.6555
	6057.5	632.4392	663.345	552.6687	607.7077	555.9561
	6154.2	650.867	672.2916	564.0046	624.1505	554.5483
	6156.7	647.9407	677.9762	567.4048	621.5232	553.7501

Quantization errors for the MFCC prove to be better than those from cepstral coefficients.

Speaker Identification

A database consisting of recordings from 2 female and 3 male subjects, ages 22-24, was created. Mel-frequency cepstral coefficients were calculated for all audio files obtained and codebooks representing each of the speakers were created from the MFCCs extracted from randomly selected speech files. These codebooks were used in efforts to identify the speaker whose voice is contained in other randomly selected audio files. Some of the data generated from the identification may be seen in Table 3.

There are 5 codebooks listed in the table representing the 5 speakers in the database. The quantization errors for each of the codebooks are recorded under their respective codebook. The MFCCs from the audio files chosen to perform the identification on are listed as the entries "Letter - melc - #".

The letters at the beginning of each MFCC entry are the same ones that have been assigned to the available codebooks. These letters represent the various speakers that form the speaker database. The codebook that results in the lowest quantization error for each of the MFCC entries is the codebook belonging to the speaker whose instance is represented by the MFCC entry. This turns out to be the case as can be seen in the table. For example, all MFCC entries beginning with B should and do have lower quantization errors under the B codebook. The same principle applies and may be observed for the remaining MFCC entries.

Table 3. Speaker Identification

	Codebook	B	L	D	N	A
	Quantization Error	5546	4530.8	5243.2	5249.6	5475.4
Audio Instance MFCCs	Bmelc1	**754.3177**	1068.7	976.3402	969.8855	898.3887
	Bmelc4	**643.9661**	1124.5	921.0814	983.8415	906.4198
	Bmelc5	**657.7202**	1097.4	921.4394	982.2822	899.7923
	Bmelc6	**781.2903**	1421.5	1059.8	1098.6	1070.8
	Bmelc15	**515.1075**	956.5698	738.5551	817.512	717.6372
	Lmelc11	971.1333	**558.1966**	1202.7	1189.3	1056.5
	Lmelc12	1079.6	**571.6959**	1320.3	1296.9	1115.7
	Lmelc13	1007.5	**549.4844**	1243.3	1248.6	1068.3
	Lmelc14	1065.6	**547.8777**	1352.2	1292.1	1117.8
	Lmelc15	1045.9	**501.0977**	1322.5	1246.5	1074.4
	Dmelc1	700.7396	1010.4	**525.9576**	749.7822	687.9495
	Dmelc3	831.0202	1191.8	**581.6313**	865.28	767.0945
	Dmelc6	660.8978	933.5458	**480.808**	695.2261	617.4146
	Dmelc9	644.9372	948.0542	**482.3799**	756.0147	668.1581
	Dmelc11	562.1126	727.2344	**423.8122**	677.5296	605.7842
	Nmelc2	969.4626	1083.1	860.8116	**638.3775**	785.4489
	Nmelc4	945.3498	1147.2	812.003	**628.4954**	723.4166
	Nmelc6	649.8373	813.9652	606.092	**452.0929**	555.8824
	Nmelc8	707.3815	754.4667	643.6351	**474.5886**	607.9611
	Nmelc11	774.9029	810.7249	725.2837	**582.2382**	711.2717
	Amelc11	735.0007	1047.5	625.5938	625.5938	**536.3636**
	Amelc12	805.5257	1092.2	726.3693	726.3693	**657.8542**
	Amelc13	874.4507	1149.6	737.7453	737.7453	**641.7227**
	Amelc14	853.4209	1168.1	684.7589	684.7589	**574.2524**
	Amelc15	835.2042	1166.8	714.1691	714.1691	**633.0814**

The quantization errors in boldface belong to the MFCC entries for audio samples corresponding to the codebook they are found under.

CONCLUSION

Text-dependent speaker identification was successfully performed with a system integrating Mel-frequency cepstral coefficient (MFCC) feature extraction and vector quantization. A database consisting of people of both sexes and varying ages was formed. The selected recording settings were WAV files with PCM format, 16 kHz sample rate. MFCC feature extraction was performed on the audio recordings for each of the speakers. Codebooks were formed from the selected training samples.

It was found that 8-bit sample size offered no clear disadvantage over 16-bit samples. Therefore 8-bit sample size was selected due to the benefits it offers with respect to performance (storage, execution times, cost). Selecting a higher size for a speaker's codebook makes it possible to attain lower quantization errors. It also allows the range of potential quantization errors to decrease and approximate some limit, making it easier to choose a proper codebook to represent a speaker.

It was demonstrated that MFCC offered not only better quantization errors for codebooks than cepstral coefficients, but also a clearer distinction during the identification process. Speaker identification performed with MFCC and vector quantization was successful and results indicate it as a feasible option for this recognition task.

APPENDIX

Florida State
UNIVERSITY

Office of the Vice President For Research
Human Subjects Committee
Tallahassee, Florida 32306-2763
(850) 644-8633 · FAX (850) 644-4392

APPROVAL MEMORANDUM

Date: 4/5/2005

To:
Jose Sanchez
188 Crenshaw Ct. #3
Tallahassee FL 32310

Dept.: **COLLEGE OF ENGINEERING**

From: **Thomas L. Jacobson, Chair**

Re: **Use of Human Subjects in Research**
Speaker Identification based on an Intergrated System Combining Cepstral Feature Extraction and Vector Quantization

The forms that you submitted to this office in regard to the use of human subjects in the proposal referenced above have been reviewed by the Human Subjects Committee at its meeting on **2/9/2005**. Your project was approved by the Committee.

The Human Subjects Committee has not evaluated your proposal for scientific merit, except to weigh the risk to the human participants and the aspects of the proposal related to potential risk and benefit. This approval does not replace any departmental or other approvals which may be required.

If the project has not been completed by **2/8/2006** you must request renewed approval for continuation of the project.

You are advised that any change in protocol in this project must be approved by resubmission of the project to the Committee for approval. Also, the principal investigator must promptly report, in writing, any unexpected problems causing risks to research subjects or others.

By copy of this memorandum, the chairman of your department and/or your major professor is reminded that he/she is responsible for being informed concerning research projects involving human subjects in the department, and should review protocols of such investigations as often as needed to insure that the project is being conducted in compliance with our institution and with DHHS regulations.

This institution has an Assurance on file with the Office for Protection from Research Risks. The Assurance Number is IRB00000446.

cc: Dr. Anke Myer-Baese
HSC No. 2005.083

Informed Consent Form

I of my own free will, without any amount of force or coercion, consent to be a participant in the research conducted for the thesis titled "Speaker Identification based on an Integrated System Combining Cepstral Feature Extraction and Vector Quantization". Jose Boris Sanchez, Masters student at the College of Engineering in the Florida State University, is performing the research. I understand that the purpose of the research is to analyze recordings of the human voice in efforts of finding accurate techniques for identifying a human by their voice.

I understand that if I decide to participate in this research I will be asked to say a common word used in everyday speech and a digital recorder will record that utterance. I understand that this will constitute one recording session. I comprehend that I will be asked to participate in 15 recording sessions, each of which is to take place no sooner than 6 hours from the previous session. Each session will take up to a minute for a total commitment time of approximately 15 minutes. I understand that the researcher and I will determine the times, dates, and place for these recording sessions in order to avoid any schedule inconvenience.

I understand that each of the recordings resulting from those sessions will be downloaded unto a computer in which the research will be performed. I understand that only the researcher will have access to the recordings at any time. I also understand that all recordings will be deleted by April 3rd, 2005. I understand that my name will not appear on any of the data resulting from the research. When making reference to the results of an individual all that will be mentioned is my gender and an assigned reference number.

I understand that the research does not involve greater than minimal risk, other than those ordinarily encountered in daily life. I understand that my participation is totally voluntary and I may stop participating at any time. I understand that no benefits or compensation has been offered to me, other than the opportunity to provide the recordings necessary for the researcher to perform his analysis and in this way contribute to the knowledge that arises from that work. I understand that this consent may be withdrawn at any time without prejudice, penalty, or loss of benefits to which I am otherwise entitled.

I have been given the right to ask and have answered any questions regarding the research. I understand that I may contact the researcher, Jose Boris Sanchez, (850)212-0090, or the researcher's major advisor, Dr. Anke Meyer-Baese at (850)410-6481 for answers to any questions regarding this research. Any questions regarding my rights as a participant may be directed to the Chair of the Human Subjects Committee, Institutional Review Board, through the Office of the Vice President for Research, (850)644-8633. I have read and understand this consent form.

______________________________ Subject

____________________ Tel.

Date ____________________

FLORIDA STATE UNIVERSITY
Approved: 4.5.05
HS No. 05.083
INSTITUTIONAL REVIEW BOARD

REFERENCES

Furui, Sadaoki. Digital Speech Processing, Synthesis, and Recognition. New York: Marcel Dekker, 2001.

Atal, B.S. “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification.” Journal of the Acoustical Society of America 55 (1974): 1304-1312.

Fant, G. “The Acoustics of Speech.” Proceedings of the Third International Conference on Acoustics 1 (1959): 188-201.

Hughes, G. and Halle, M. “Acoustic Properties of Stop Consonants.” Journal of the Acoustical Society of America 30 (1957): 107-116.

Atal, B.S. “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification.” Journal of the Acoustical Society of America 55 (1974): 1304-1312.

Fujimura, O. “Analysis of nasal consonants.” Journal of the Acoustical Society of America 34 (1962): 1865-1875.

Blumstein, S. and Stevens, K. “Perceptual invariance and onset spectra for stop consonants in different vowel environments.” Journal of the Acoustical Society of America 67 (1980): 648-662.

Blumstein, S. and Stevens, K. “Invariant cues for place of articulation in stop consonants.” Journal of the Acoustical Society of America 64 (1978): 1358-1368.

Itakura, F. and Saito, S. “Speech information compression based on the maximum likelihood spectrum estimation.” Journal of the Acoustical Society of Japan 27 (1971): 463-470.

Tokhura, Y. “A weighted cepstral distance measure for speech recognition.” IEEE Transactions on acoustics, speech and signal processing 35 (1987): 1414-1422.

Schafer, R. and Rabiner, L. “Systems for Automatic Formant Analysis of Voiced Speech.” Journal of the Acoustical Society of America 47 (1970): 634-648.

Schafer, R. and Rabiner, L. “Digital Representation of Speech Signals.” Proceedings of the IEEE 63 (1975): 662-677.

Gray, R.M. “Vector Quantization.” IEEE ASSP Magazine 1 (1984): 4-29.

BIOGRAPHICAL SKETCH

Jose Boris Sanchez began studies at Florida State University in the fall of 1998 and completed studies in the Electrical and Computer Engineering program in December of 2002. He was accepted into graduate school by the Department of Electrical and Computer Engineering and began studies in the pursuit of a Master of Science in Electrical Engineering during the spring semester of 2003. Pending approval and successful defense of the work contained in the thesis, successful completion of the current degree is expected during the spring semester of 2005.