Speaker Sound Coding Using Vector Quantization Technique (Vq)

Oday Kamil Hamid
Computer Techniques Engineering Department, Dijla University College/Baghdad
Email: uday10feb@yahoo.com

Received on: 12/11/2012 & Accepted on: 5/9/2013

ABSTRACT

The key objective of this research is to compress the speech sound or in another meaning coding speaker sound with a small matrix called code book represent the speaker sound information this is done by using Vector Quantization method (VQ).The sound features here represented with both types Linear Predictive Coding (LPC) and autocorrelation coefficients that represent the data base of this research.

The algorithm is tested upon a database consist of matrix with dimensions (54*13) which represent speaker sound information with normalized autocorrelation coefficients and compress it to a codebook (CB) contain 8 codeword (8 CW) with autocorrelation coefficients and the other CB with LPC coefficients of order p=12. From this project we can notes how much the sound information can be compressed which represented at first with a matrix of dimensions (54*13) and transfer it to a matrix with dimensions (8*12) or (8*13) with same feature sound of speaker.

In this project there is no focus on LPC method and how it work but used it in extracting Sound features in another program which represent data base for this project and inter it to VQ algorithm

The algorithm was examined through computer simulation using Matlab version 6 programming language and under Microsoft Windows XP operating system.

Keywords: Autocorrelation Coefficients, Linear Predictive Coding, Codebook.

ترميز صوت المتكلم باستخدام تقنية تكميم المتجهات

الخلاصة

ان الهدف الرئيسي لهذا البحث هو ضغط المعلومات الصوتية للكلام اوبمعنى اخر تشفير صوت المتكلم بمصفوفة صغيرة تدعى كتاب التشفير يمثل معلومات صوت المتكلم وهذا تم تنفيذه باستخدام طريقة تكميم المتجهات VQ. ان الميزات الصوتية هنا تم تمثيلهابكلا النوعين، معامل تشفير التنبؤ الخطي ومعامل الارتباط.

ان الخوارزمية تم اختبارها على قاعدة بيانات تتكون من مصفوفة ابعادها (54*13) والتي تمثل المعلومات الصوتية للمتكلم بمعامل الارتباط الطبيعية وضغطها لكتاب تشفير CB يتكون من 8 كلمات تشفير CW مع معامل الارتباط الطبيعي والكتاب الاخر مع معامل تشفير التنبؤ الخطي

https://doi.org/10.30684/etj.32.6A.5
2412-0758/University of Technology-Iraq, Baghdad, Iraq

مع رتبة تشفير P=12.ومن خلال هذا البحث نلاحظ كم تم ضغط المعلومات الصوتية حيث كانت تتمثل بمصفوفة حجمها (13*54) وتحويلها الى مصفوفه حجمها (12*8) او (13*8) مع بقاء الميزه الصوتيه للمتكلم.

في هذا البحث لم يتم التركيز على طريقة تشفير التنبؤ الخطي وكيفية عملها ولكن تم استخدمتها في برنامج اخر لاستخراج الميزات الصوتية لم يتم التطرق اليه واعتبارها كقاعدة بيانات لهذا البحث وادخالها لخوارزمية تكميم المتجهات VQ .

ان الطريقه المقترحة قد تم اختبارها من خلال برامج محاكاة بالحاسبة باستخدام برنامج Matlab 6 كلغة برمجة تحت نظام تشغيل وهو XP.

INTRODUCTION

The theory of vector quantization (VQ) has established a wide variety of techniques for quantization spectral shape to minimize overall spectral distortion. Such vector quantizers have been widely used in the areas of speech coding, vocoding, image coding and speech & speaker recognition [1, 2].

Vector quantization is a data coding technique in which the vector of acoustic parameters, which represents a segment of speech, is replaced by a single codeword. This codeword is an index number to an entry in a predetermined ordered set of reference segments referred to as codebook. It represents a partitioning of acoustic space in the domain of speech being quantized [3].

Vector quantization (VQ) is a generalization of scalar quantization to the quantization of a vector (an ordered set of real numbers). VQ has received much attention as a powerful technique for data compression. It has been shown that VQ is an effective method for coding speech, images, and video signals. According to Shannon's rate-distortion theory, a better performance is always achievable in theory by coding vectors instead of scalars, even though the data source is memory less. Furthermore, the vector extension of Bennett's high quantization theory shows explicitly the performance gain that may be achieved by using VQ's. There exists major advantages of a VQ over a scalar quantize.

THEORY

The basic concept of vector quantization is schematically depicted in figurer (1) and it is constructed as: -

The speech is segmented into successive short frames and a vector of finite dimensionality represents each frame of speech. The vector may be in the form of sampled data or autocorrelation coefficients terms or linear predictive coefficients, etc, this vector is called training vector [4].

These vectors that produced from last step are mapping from 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 codeword (CW), which represents feature vector for a known speaker. The collections of all codewords are called codebook (CB) [5].

Codebook generation requires an iterative process much like a clustering algorithm involving a large number of spectral comparisons, and produces a finite collection of spectral model vectors (codebook) so that the average spectral distortion from all the input vectors to their best match in the vector collection is minimized. The same spectral comparison strategy in the codebook generation process is executed in the quantizers. Each input speech vector is mapped to the codebook entry (codeword) index corresponding to the best match vector. Speech compression or rate reduction is accomplished by using the indexes as storage or transmission parameters [4].

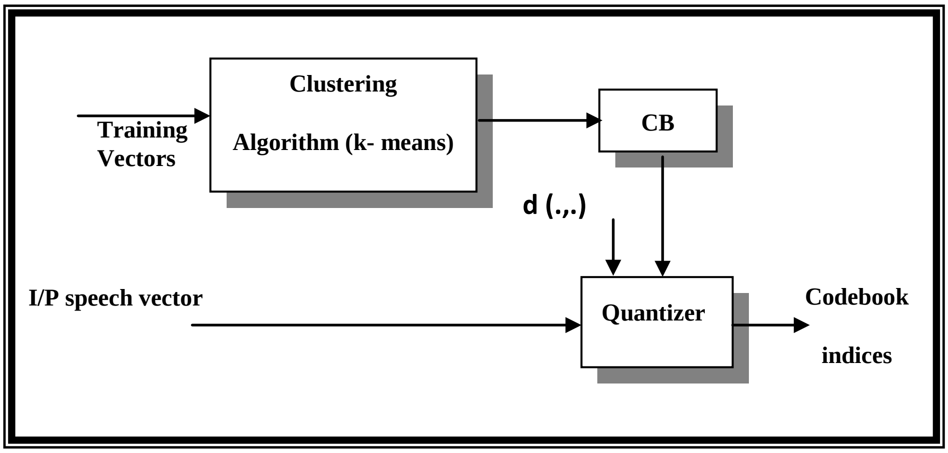

Figure (1): -Block diagram of the basic VQ training and classification structure

Hence Vector quantization technique can be regarded as a classification or clustering method where multidimensional space occupied by the training vectors is partitioned into a set of mutually exclusive convex regions each of these regions represents a different cluster of data i.e. code words. The collection of centroids of the regions will represent the codebook as shown in figure (2). So it is possible to classify input vectors into groups that have the same features in a single group [6,7].

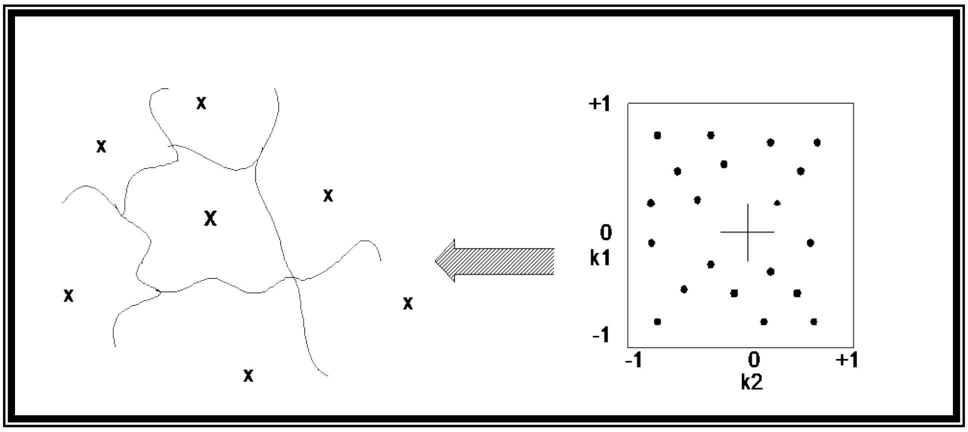

Figure (2): Partitioning of a vector space into VQ cells with each cell represented by a centroid vector.

Vector quantization (VQ) can perform data compression technique in which vector of acoustic parameters, that represents a segment of speech, can be replaced by a single symbol or codeword. This symbol is an index number to an entry in a pre-determined ordered set of reference segments, referred to as the codebook; these codebooks represent a partitioning of the acoustic space in the domain of the speech being quantized [8].

Thus Vector quantization provides the capability of mapping vectors from a large vector space to a finite number of regions in that space .The data is thus significantly compressed; yet still accurately represented. Without quantizing the feature vectors the system would be too large and computationally complex. In a speaker recognition system, the vector space contains a speaker's characteristic vectors, which are obtained from the feature extraction, which will describe latterly. After vector quantization takes place, only a few representative vectors remain, collectively known as delineation for the speaker, and is used when training a speaker in the system [9].

Advantages and Disadvantages of VQ

The key advantages and disadvantages of the VQ representation are [6]: -

1. Reduced storage for spectral analysis information (data compression).
2. Reduced computation for determining similarity of spectral analysis vectors. exist that exploit these labels so as to recognize speech in an efficient manner.

The disadvantages of using a VQ codebook to represent speech spectral vectors are: -

1. An inherent spectral distortion in representing the actual analysis vector.
2. The storage required for codebook vectors is often nontrivial. The larger we make the codebook (so as to reduce quantization error); the more storage is required for the codebook entries.

Codebook (CB) Generation

Codebook is composed of different vectors, which represent the essential characteristics of each speaker. The way in which a set of vectors called codeword (CW) which represents codebook and it is equal to M but this condition must be achieved [10]: -

$$D(M) = \min_{(M)}\left[\frac{1}{L}\sum_{i=1}^{L}\min_{1 \le m \le M}\left(d\left(v_i, cw_m\right)\right)\right] \qquad ..(1)$$

Where

M: Codebook size (number of code words).

D(M) = Average distortion

$d\left(v_i, cw_m\right)$ = distance between training vectors and codeword vectors

Codebook generation process can be represented in a well-known algorithm, namely *LGB* algorithm (Linde, Gray and Buzo) and it is also named by k-means algorithm or vector quantization (VQ) algorithm and these steps represent it [11]:

a. Initialization

Arbitrarily choose *M* vectors (initially out of the training set of *L* vectors) as the initial set of code words in the codebook, these training vectors may be in autocorrelation coefficients form or reflection coefficients.

b. Finding nearest neighbor

For each training vector, find the codeword in the current codebook that is closest by using any method of distance measure described later, and assign that vector to the corresponding cell (associated with the closest codeword) and then total average distance computed as: -

$$D = \frac{1}{L}\sum_{i=1}^{L} \min_{1 \le m \le M}\left(d\left(v_i, cw_m\right)\right) \qquad \text{….. (2)}$$

c. Centroid Update:

Update the codeword in each cell using the centroid of the training vectors assigned to that cell corresponding to this equation:

$$Y_j = \frac{1}{n}\sum_{i=1}^{n} v_{ij} \qquad \text{….. (3)}$$

where

v_{ij} = The ith vector in the jth cluster.

Y_j = The jth new codeword.

n= Number of vectors in each cluster.

Iteration:

Repeat step 2&3 until the average distance falls below a preset threshold. The k-means algorithm that represented by previous steps give us a good results in codebook generation but there is two point must be taken into account, binary split-algorithm and manipulate empty (bin) cell.

Binary Split Algorithm

Although the above iterative procedure works well, it has been shown that it is advantageous to design an M-vector codebook (CB=Mcw) in stages i.e., by first designing a 1-vector codebook, then using a splitting technique on the code words to initialize the search for a 2-vector codebook, and continuing the splitting process until the desired M-vector codebook is obtained. This procedure is formally implemented by the following procedure and flow diagram of figure (3) [6]: -

1. Design a 1-vector codebook; this is the centroid of the entire set of training vectors (hence, no iteration is required here).

2. Double the size of codebook by splitting each current codebook y_m according to the rule:

$$y_m^+ = y_m(1+\varepsilon) \qquad \text{…(4)}$$

$$y_m^- = y_m(1-\varepsilon) \qquad \text{…(5)}$$

y_m^+, y_m^- = Codebook vectors after splitting.

y_m = Codebook vectors before splitting.

Where *(m)* varies from 1 to the current size of the codebook *(M)*

$1 \le m \le M$

$\varepsilon =$ Splitting parameter $0.01 \le \varepsilon \le 0.05$ (ε =0.01 used in this algorithm)

3. Use the k-means iterative algorithm (as discussed above) to get the best set of centroids for the split codebook.
4. Iterate steps 2 and 3 until a codebook of size *M* is designed.

Empty (bin) Cell

One important point that we must take it into account its to verify if there is an empty bins and manipulate it if it is available, these empty bins produced from generation codebook of large size from small training vectors. And it can be implemented by the following procedure [1, 12]: -

1. Find the most popular cell, which is the one that possess the highest number of training vectors that are classified to its codeword.
2. Form two new code words whose initial codeword is obtained by multiplying the codeword of the most popular cell by 1.01 and 0.99, then delete the previous popular codeword and the empty cell codeword.
3. If there are more empty cells, then find the second most popular cell and replace its codeword together with the second empty cell codeword.

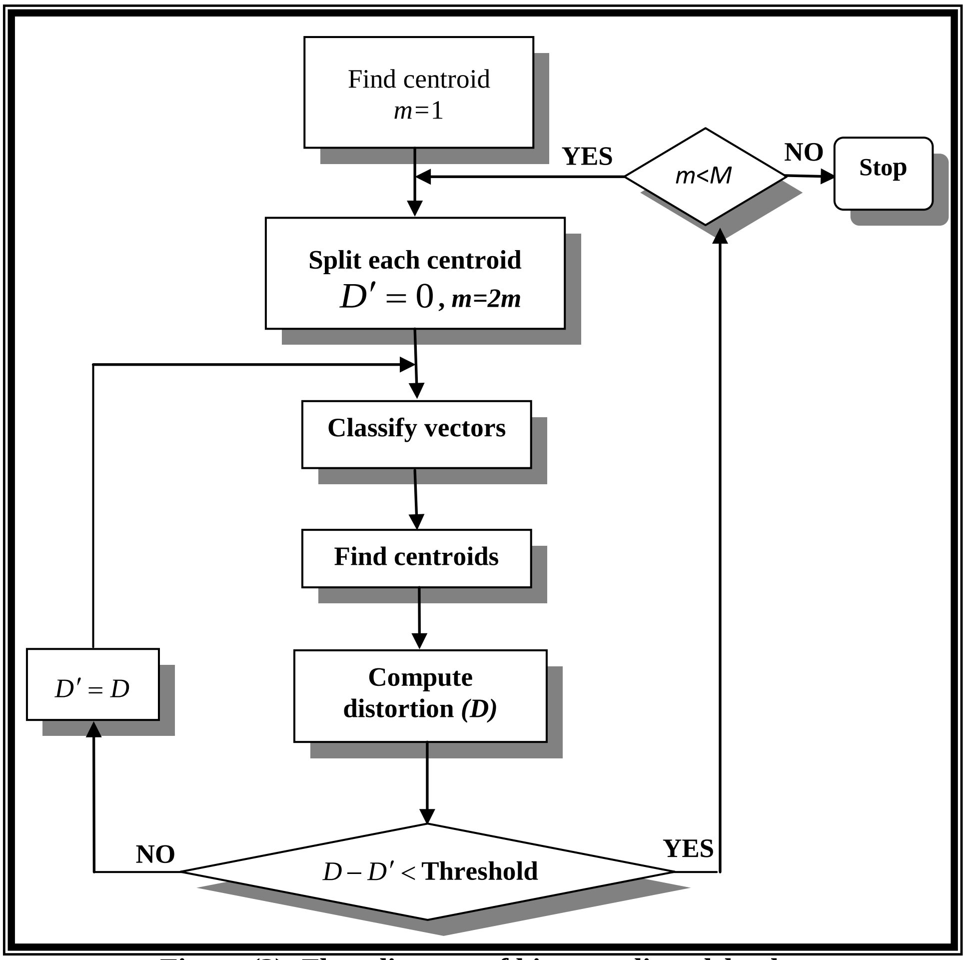

Figure (3): Flow diagram of binary split codebook Generation algorithm.

PRACTICAL WORK

In this chapter the results of proposed algorithm will be introduced Before that the theory of vector quantization method as dependent on many studies they where find that if the size of code book increase the distortion measure decrease but here in our research we use code book with 4 codeword regardless of distortion measures.

The data base of our work is a matrix with dimension is (54*13); represent the feature sound of one male speaker with autocorrelation coefficients

The matrix is interred to the vector quantization method to generate a code book with 4 code word size with LPC coefficient and second step interred the same matrix is interred to the same algorithm and generate the same size of code book but with autocorrelation coefficients.

Chapter four introduces the practical results for the project.

Codebook Generation Algorithm

In this phase VQ was used in order to build a reference for a speaker in the system by clustering the feature vectors in k-separate clusters. A codeword vector then represents each cluster; the resulting set of codeword vectors is called a codebook. Thus, the distribution of the feature vectors is represented by a smaller set of sample vectors with similar distribution than the full set of feature vectors of the speaker model. Figure (4) illustrates how centroids can represent speakers efficiently.

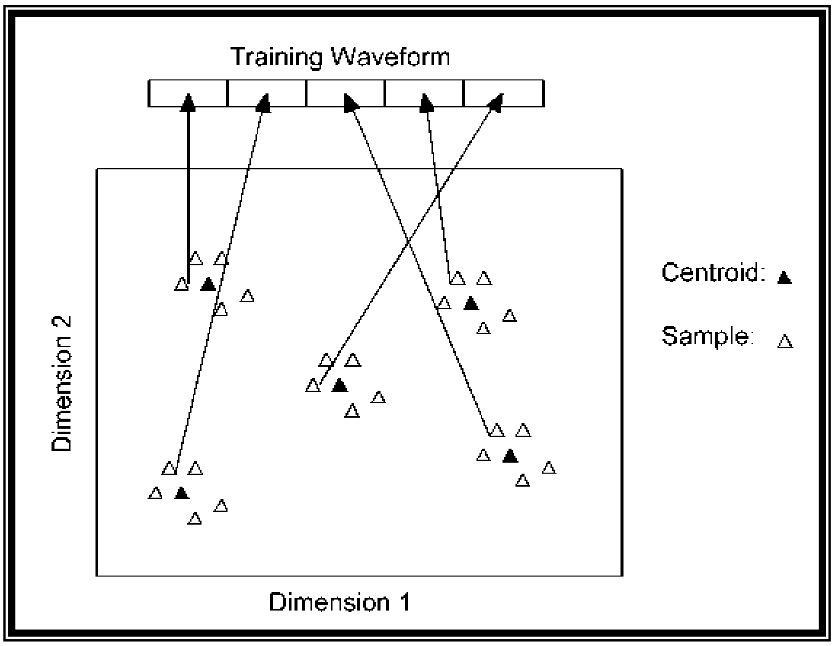

Figure (4): Conceptual diagram illustrating codebook information.

The steps of this program will be explained briefly as in flowchart of figure (5). And during the current research the researcher used the generation program in order to generate codebooks with different sizes to use them in main programs as the system needs, and the size of input data, and number of speaker in the system. But the main goal of all these programs is to quantize input vectors and compress them during repetitive steps where average distortion decreases with the size of code book and get on a smallest distortion for codebook size we want.

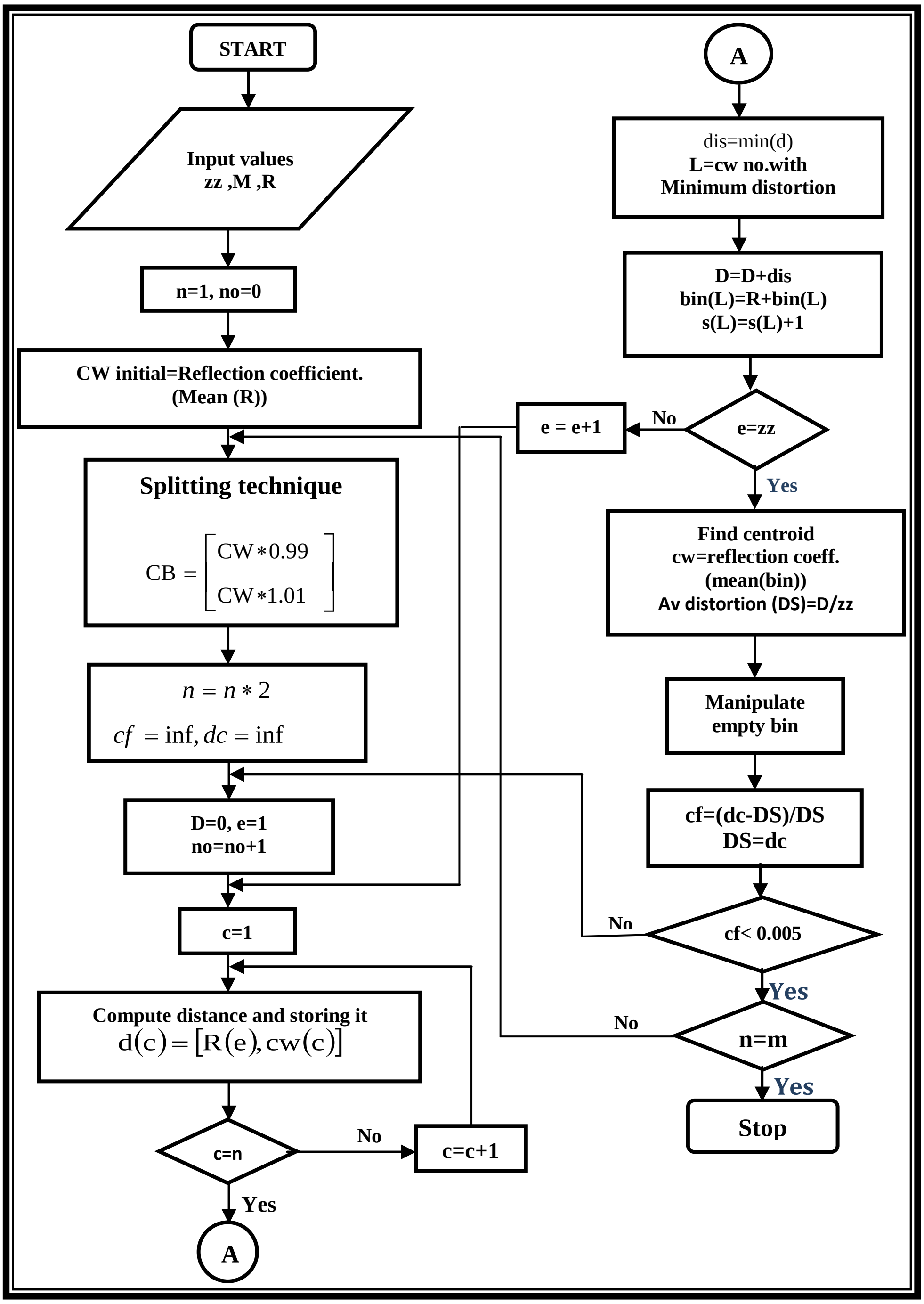

Figure (5): Flowchart of codebook generation program.

The list of abbreviation in figure (5) listed below:
zz = number of features vectors.
M = codebook size.
R = Features vectors group in autocorrelation coefficients.
no = Number of repetition.
cf = Threshold value that limited number of repetition.
D = Total distance between features vectors and current codebook.
n = Size of current codebook.
bin = cells that contained classified feature vectors.
S = Number of features vectors in each bin.
L = Codeword number recorded minimum distance with feature vector.
Ds = Average distortion result from quantization feature vectors with Respect to current codebook.

RESULTS

Codebook generation program was used to generate codebooks with size 8 CW. Table (1) shows the results of the program shown on computer screen in autocorrelation coefficients with size (8*12) and table (2) with linear predictive coefficients its size (8*13) .

Where a codebook with size (8CW) was generated for a known speaker in three stages by using (k-means) algorithm, tables (3), (4), and (5) shows the values of average distortion (AV. Distortion) and number of feature vectors in each cell (c1, c2…), cf is the threshold value that limits the number of iterations. Figure (7) represent spectral shape for this codewords. The precedent studies proved that codebook size is wealthy in increasing the recognition ascription with regardless for the algorithm used to generate it [13]. So the researcher generates different sizes of codebooks for one speaker.

Table (6) and figure (6) describe the inverse relation between average distortion and codebook size.

Table (1): Codebook with Autocorrelation Coefficients

-0.9277	0.5428	0.0562	0.2281	0.0548	0.0158	0.0390	-0.1136	-0.2156	-0.0545	0.1799	0.1913
-0.9302	0.3836	0.0799	0.4121	0.1569	0.0504	-0.0570	0.0218	-0.1445	-0.1193	0.1408	0.1708
-0.9570	0.5650	0.2637	0.4367	0.1343	-0.0466	-0.0152	-0.0025	-0.0312	0.0171	-0.0581	-0.1464
-0.9749	0.5451	0.5254	0.2619	0.1200	-0.1416	-0.0917	0.0175	-0.1705	-0.0501	0.1450	0.1127
-0.0793	-0.5528	-0.4219	-0.3259	-0.0949	-0.0046	0.0772	-0.0096	-0.1453	-0.1018	-0.0699	0.0340
-0.0813	-0.5829	-0.4880	-0.5012	-0.0421	-0.1567	0.1271	-0.0376	-0.0607	0.0011	0.0535	0.0546
0.4460	-0.3197	-0.4293	-0.3072	-0.3459	-0.0702	-0.1217	-0.2144	-0.0172	-0.0442	0.0746	0.0841
0.2858	-0.4256	-0.4259	-0.5737	-0.4049	-0.1100	-0.0702	0.0117	0.1551	-0.1727	0.1696	0.1251

Table (2): Codebook with linear predictive coefficients

1.0000	-1.3176	0.4273	-0.1998	0.1573	0.0594	-0.0569	0.1147	0.1419	-0.0866	-0.2034	-0.0787	0.1913
1.0000	-1.1337	0.2765	-0.3811	0.2370	0.0415	0.1821	-0.0388	0.1275	-0.0268	-0.2246	-0.0570	0.1708
1.0000	-1.1733	0.1767	-0.2758	0.2525	0.1882	-0.0225	-0.0647	0.0159	-0.0205	0.0580	0.1149	-0.1464
1.0000	-1.0418	-0.3612	0.1577	0.1323	0.2414	-0.0451	-0.0165	0.2462	-0.1441	-0.2407	0.0257	0.1127
1.0000	0.3865	-0.3183	-0.4958	-0.3918	-0.0567	0.1421	0.1938	-0.0077	-0.1764	-0.1391	-0.0567	0.0340
1.0000	0.5033	-0.1954	-0.5655	-0.5173	-0.0897	-0.0573	0.0861	-0.1259	-0.1012	0.0171	0.0808	0.0546
1.0000	0.7452	0.0123	-0.2884	-0.3787	-0.3201	-0.1625	-0.3120	-0.2773	-0.0727	0.0122	0.1368	0.0841
1.0000	0.8405	0.1260	-0.4244	-0.8179	-0.5498	-0.1953	-0.1644	-0.0585	-0.0277	-0.0126	0.2721	0.1251

Table (3): Results of First Stage In Codebook Generation Program

ITERATION NO.	AVERAGE DISTORTION	CF VALUE	VECTORS/BIN	
			C1	C2
1	1.5527	Inf	19	35
2	1.1726	0.3242	40	14
3	0.3921	1.9902	41	13
4	0.3902	0.0050	41	13
5	0.3902	0	41	13

Table (4): Results of Second Stage In Codebook Generation Program

ITERATION NO.	AVERAGE DISTORTION	CF VALUE	VECTORS/BIN			
			C1	C2	C3	C4
1	0.3592	Inf	15	26	5	8
2	0.2996	0.1991	12	29	5	8
3	0.2945	0.0172	11	30	5	8
4	0.2942	0.0011	11	30	5	8

Table (5): Results of third Stage In Codebook Generation Program

ITERATION NO.	Average Distortion	CF VALUE	VECTORS/BIN							
			C1	C2	C3	C4	C5	C6	C7	C8
1	0.2748	Inf	6	5	22	8	2	3	4	4
2	0.2239	0.2272	7	5	21	9	2	3	4	4
3	0.2155	0.0392	7	4	21	9	2	3	4	4
4	0.2095	0	7	4	21	9	2	3	4	4

Table (6): Relationship Between Codebook Size and Average Distortion

CB SIZE		2	4	8	16	32	64
Sp	Iteration no.	5	3	5	4	5	5
	Av. distortion	0.3597	0.2558	0.1671	0.0955	0.0314	0.0281

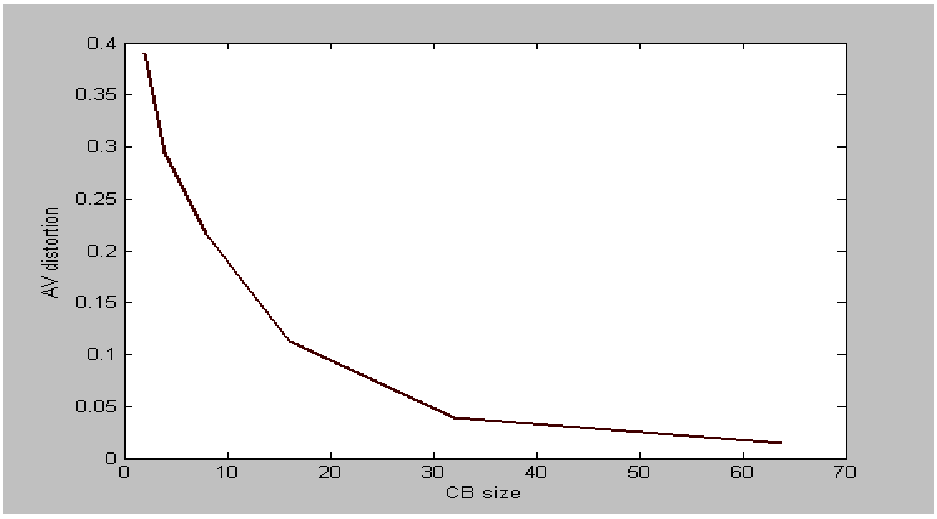

Figure (6): Relation Between Codebook Size and Average Distortion

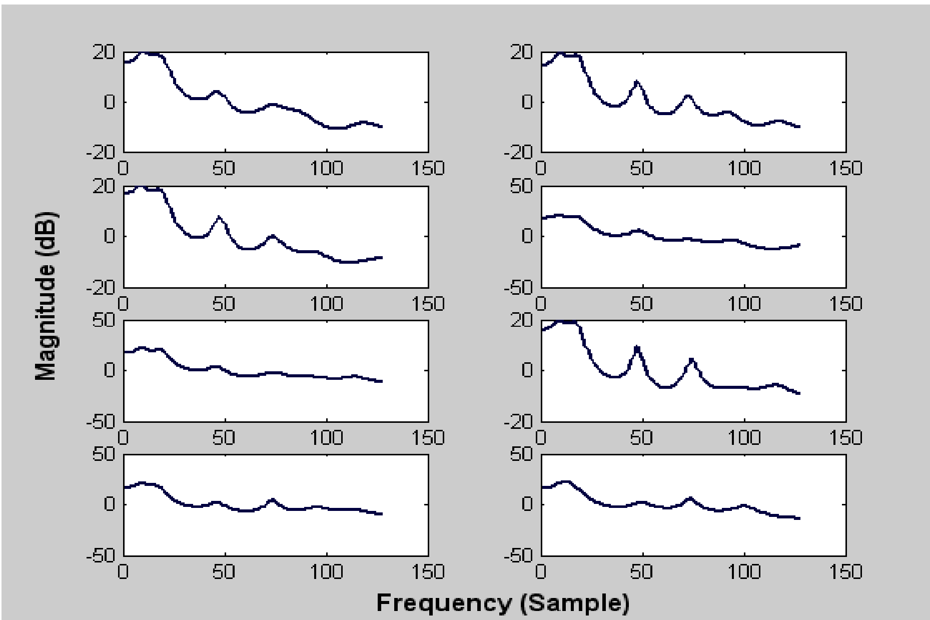

Figure (7): Spectral Shape for the Eight Codewords in the Main Codebook

CONCLUSIONS

1) The efficiency and effectiveness of the VQ technique in building system and compressing data.
2) The employment of the programming language (Matlab) becomes wealthy in the shorthand on the program volume and the ease of its correction.

Recommendations for future work

1) Using another teqnique like Tree-Structured VQ (TSVQ) or Product-Code VQ (PCVQ).
2) Use another parameter instead of LPC like log area ratio coefficients, cepstral coefficients.
3) Using VQ in image processing.

REFERENCES

[1]Al-Nakkash, A. H., “A Novel Approach For Speakers Recognition Using Vector Quantization Technique”, M.Sc. Thesis, University of Technology, Baghdad, 2001.

[2]Rabinar, L. R., Sondhi, M. M., and Levinson, S. E., “A Vector Quantization Combining Energy and LPC Parameters and Its Application To Isolated Word Recognition”, AT&T Bel Laboratories Technical Journal, Vol.63,No.5, June 1984.

[3]Bashik, S. N., “A Proposed Arabic Phoneme Segmentation Algorithm”, M.Sc. Thesis, College of Engineering, University of Baghdad, 2001.L. R. Rabinar and B. H.Juang, “Fundamentals of Speech Recognition”, Prentice-Hell, New Jersey, 1993.

[4]Juang, B. H., Wong, D. Y., and Gray, A. H., “ Distortion Performance of Vector Quantization For LPC Voice Coding”, IEEE Trans. Acoust. , Speech and Signal Process. , Vol. ASSP-30, April 1982, pp.294-303.

[5] Minh, N. Do., “An Automatic Speaker Recognition System”, Swiss Federal Institute of Technology, lausanne-Epel, “http://lcavwww. Epfl. Ch./~mindho/asr_project/. Html”, January 2000.

[6]Rabinar, L. R. and Juang, B. H., “Fundamentals of Speech Recognition”, Prentice-Hell, New Jersey, 1993.

[7]Ganesh, M. and Sugumaran, K., “Speaker Verification In Forensic Application”, Surprise 95 Journals, Vol. 4, Imperial Collage of Science Technology & Medicine, London, 1995.

[8]Mahmoud, W. A., “An Automatic Algorithm For Creating Reference Patterns”, Digital Signal Processing Symposium, Kuwait, 1990.

[9]Boettcher, P., Siegendorf, J. and Soong, T. S, “Text-Independent Speaker Identification”, http:/www.ece.cmu.edu/~ee551/old_projects/s_98_10/proposel. Html”, spring 1998.

[10]Furui, S. F., “Cepstral Analysis Technique For Automatic Speaker Verification”, IEEE Trans. Acoust. Speech, Signal Processing, 29, 254-272, 1981.

[11]AL-Nakkash, A. H., Mahmoud, W. A. and AL-Sadik, A. E., “Text-independent speaker recognition based on VQ techniques”, The Blister of The Engineering, Vol. 7,No. 2, 2001.

[12]Mahmoud, W. A., “Quantization Techniques for The Classification and Recognition of Speech Signals”,Ph.D.Thesis,University of Swansea,England,1986.

[12]Oglesby, J. and Mason, J. S., “Opti mization of Neural Models for Speaker Identification”, Proc. ICASSP, 90. 261-264, 1990.