International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:3, 2008

Speech Data Compression using Vector Quantization

H. B. Kekre, Tanuja K. Sarode

Abstract**—**Mostly transforms are used for speech data compressions which are lossy algorithms. Such algorithms are tolerable for speech data compression since the loss in quality is not perceived by the human ear. However the vector quantization (VQ) has a potential to give more data compression maintaining the same quality. In this paper we propose speech data compression algorithm using vector quantization technique. We have used VQ algorithms LBG, KPE and FCG. The results table shows computational complexity of these three algorithms. Here we have introduced a new performance parameter Average Fractional Change in Speech Sample (AFCSS). Our FCG algorithm gives far better performance considering mean absolute error, AFCSS and complexity as compared to others.

Keywords**—**Vector Quantization, Data Compression, Encoding,, Speech coding.

I. INTRODUCTION

THANKS for the rapid technological growth and the usage of the internet today has made possible for the transmission of multimedia applications over the web. The multimedia applications consist of mainly speech, images, and videos. These applications requires large amount of data resulting in consumption of huge bandwidth and storage resources. Vector quantization (VQ) [1]-[3] is an efficient technique for data compression and has been successfully used in various applications involving VQ-based encoding and VQ-based recognition. The response time is very important factor for real time application [1]. Many type of VQ, such as classified VQ [9], [10], address VQ[9], [11], finite state VQ[9], [12], side match VQ[9], [13], mean-removed classified VQ[9], [14], and predictive classified VQ[9], [15], have been used for various purpose. VQ has been applied to some other applications, such as index compression [9], [16], and inverse half toning [9], [17], [18]. VQ has been very popular in a variety of research fields such as speech recognition and face detection [5], [19], pattern recognition [22]. VQ is also used in real time applications such as real time video-based event detection [5], [20] and anomaly intrusion detection systems [5], [21].

VQ can be defined as a mapping function that maps k-dimensional vector space to a finite set **CB = {C_1, C_2, C_3, ……, C_N}**. The set CB is called codebook consisting of N number of codevectors and each codevector $C_i = \{c_{i1}, c_{i2}, c_{i3}, \ldots\ldots, c_{ik}\}$ is of dimension k. The key to VQ is the good codebook. Codebook can be generated in spatial domain by clustering algorithms or using transform domain techniques [6]-[8]. The method most commonly used to generate codebook is the Linde-Buzo-Gray (LBG) algorithm [3], [4] which is also called as Generalized Lloyd Algorithm (GLA).

Speech coding techniques are mostly based on lossy algorithms. Lossy algorithms are considered acceptable when encoding speech because the loss in quality is undetectable by the human ear. Uncompressed speech is usually transmitted at 64 kb/s, using 8 bits/sample and at a rate of 8 KHZ for sampling. Any bit rate below 64 kb/s is considered as compression. There are many lossy compression technique present in literature called Linear predictive coding [24]-[26], Multi stage vector quantization technique (MSVQ) [27]-[29], Switched split vector quantization technique (SSVQ) [27], Multi switched split vector quantization (MSSVQ) [23]. In this paper we propose speech coding techniques using VQ algorithms LBG, Kekre's Proportionate Error KPE[30]-[32] and Kekre's Fast codebook Generation FCG[31]. Here we have generated codebooks of size 256 for six different speech samples considering vector dimension 16 and 32.

In the next section we present VQ algorithms LBG, KPE, and FCG. In section III consist of results and conclusions in section IV.

II. CODEBOOK GENERATION ALGORITHMS

A. LBG Algorithm [3], [4]

In this algorithm centroid is computed as the first codevector for the training set. In Fig. 1 two vectors v_1 & v_2 are generated by adding constant error to the codevector. Euclidean distances of all the training vectors are computed with vectors v_1 & v_2 and two clusters are formed based on nearest of v_1 or v_2. This procedure is repeated for every cluster. The drawback of this algorithm is that the cluster elongation is $+135^o$ to horizontal axis in two dimensional cases. This results in inefficient clustering.

Dr. H. B. Kekre is Senior Professor working with Mukesh Patel School of Technology, Management and Engineering, NMIMS University, Vile-Parle (West), Mumbai-56. (E-mail: hbkekre@yahoo.com)

Ms. Tanuja K. Sarode, is Ph.D. Scholar from Mukesh Patel School of Technology, Management and Engineering, NMIMS University, Vile-Parle (West), Mumbai-56. Assistant Professor working with Thadomal Shahani Engineering College, Bandra (West), Mumbai-50. (E-mail: tanuja_0123@yahoo.com).

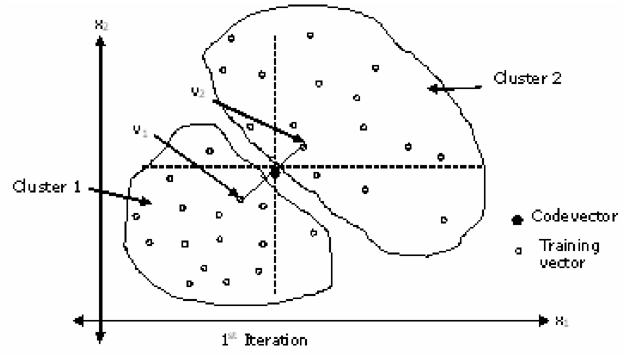

Fig. 1. LBG for 2 dimensional case

B. Proportionate Error Algorithm (KPE) [30]- [32]

Here proportionate error is added to the centroid to generate two vectors v_1 & v_2. Magnitude of members of the centroid decides the error ratio. Hereafter the procedure is same as that of LBG. While adding proportionate error a safe guard is also introduced so that neither v_1 nor v_2 go beyond the training vector space. This removes the disadvantage of the LBG.

Both LBG and KPE requires 2M number of Euclidean distance computations and 2M number of comparisons where M is the total number of training vectors in every iteration to generate clusters.

C. Fast Codebook Generation Algorithm (FCG)

We have proposed FCG for image data compression [32]. In this paper we are proposing this algorithm for speech data compression along with LBG and KPE and comparative performance of these algorithms is given.

Let $\mathbf{T= \{X_1, X_2, \ldots\ldots, X_M\}}$ be the training sequence consisting of **M** source vector. Assume that source vector is of length **K**, $\mathbf{X_m=\{x_{m,1}, x_{m,2}, \ldots\ldots x_{m,K}\}}$ for **m=1,2, …., M.** Let **N** be the number of codevectors and let C = $\{c_1, c_2, \ldots\ldots\ldots, c_N\}$,represents the codebook. Each codevector is k dimensional,

e.g., $\mathbf{c_n = (c_{n,1}, c_{n,2}, \ldots\ldots\ldots, c_{n,k}), n= 1, 2, \ldots, N.}$

In this algorithm codebook is generated based on comparison technique and hence is faster as compared to LBG and KPE. Initially we have only one cluster, centroid of this cluster is computed and then this cluster is split into two parts by comparing the first element of all the vectors present in the cluster with the first element of the centriod.

For all $x_{i,1}$ where i = 1, 2, ….. , M if $x_{i,1} < c_{1,1}$ then $x_{i,1}$ is grouped into cluster 1 or else $x_{i,1}$ is grouped into cluster 2. Centroid of cluster 1 (i.e. c_1) and cluster 2 (i.e. c_2) is computed and cluster 1 is split into two by comparing the second element of all the vectors present in the cluster 1 with the second element of the centroid c_1. Similarly cluster 2 is split into two by comparing the second element of all the vectors present in the cluster 2 with the second element of the centroid c_2. Now four clusters are formed centroids of all these clusters are computed and each of these cluster is split further by comparing the third element of all vectors with the third element of the centroid. The process is repeated further till the codebook of desire size is obtained. The above process is depicted in Fig. 2 for two dimensional case.

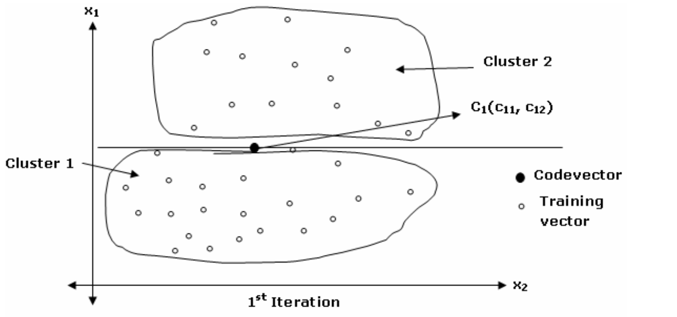

Fig. 2a

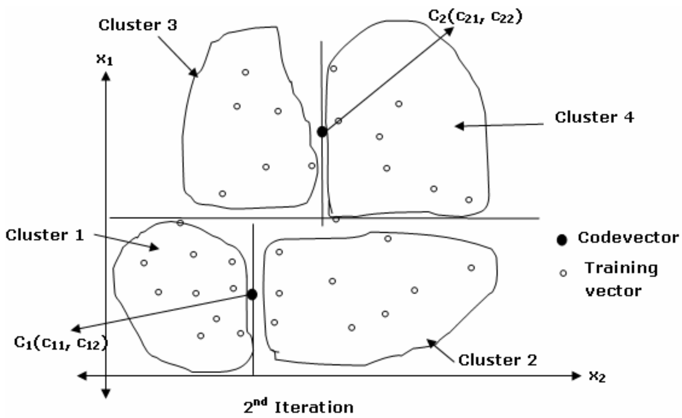

Fig. 2b

Fig. 2 FCG algorithm for 2 dimensional case

It is observed from the results that this algorithm gives minimum error and also least time to generate codebook as compared to all other clustering algorithm i.e LBG and KPE. This algorithm requires only comparisons for generating clusters. For M training vectors we require M comparisons in each iteration.

III. RESULTS

VQ has many applications in this paper we have chosen speech compression as an application. The algorithms are implemented using Pentium IV 1.7 GHz 512 MB RAM, using Matlab 6.

Here we have introduced a new performance parameter which is named as Average Fractional Change in Speech Sample (AFCSS) and it is computed as follows:

$$\frac{1}{P}\sum_{P}\frac{\left|f(x)-\hat{f}(x)\right|}{f(x)} \tag{1}$$

Where $f(x)$ is original speech signal of size P and $\hat{f}(x)$ is the reconstructed speech signal

Table 1 shows the comparison of LBG, KPE, FCG with respect to Average fractional change in speech sample (AFCSS) and Mean Absolute Error for six different speech signals.

Table 2 shows the comparison of LBG, KPE and FCG algorithm with respect to total number of Comparisons, total number of ED computations and total CPU units required.

Using formulae given in table 2, table 3 gives the total CPU units required for all six speech samples.Table 3 shows Total CPU units required for all speech samples.

TABLE I COMPARISON OF LBG, KPE, AND FCG ALGORITHMS WITH RESPECT TO AVERAGE FRACTIONAL CHANGE IN SPEECH SAMPLE AFCSS AND MEAN ABSOLUTE ERROR FOR SIX DIFFERENT SPEECH SIGNALS

Speakers	Performance Criteria	Codebook Size 256								
		Vector Dimension 8			Vector Dimension 16			Vector Dimension 32		
		FCG	LBG	KPE	FCG	LBG	KPE	FCG	LBG	KPE
HBK	AFCSS	0.6052	0.7294	0.7291	0.7585	0.8919	0.8912	0.9148	1.1378	1.0196
	MAE	0.0012	0.0018	0.0018	0.0017	0.0024	0.0024	0.0021	0.0027	0.0027
Tanuja	AFCSS	0.7657	0.9916	0.9924	1.0602	1.1280	1.1120	1.3207	1.5167	1.4275
	MAE	0.0028	0.0041	0.0041	0.0041	0.0052	0.0051	0.005	0.0064	0.0058
Sudeep	AFCSS	0.9070	1.0434	1.0247	1.0991	1.2329	1.2335	1.1534	1.2022	1.2069
	MAE	0.0039	0.0055	0.0054	0.0047	0.0066	0.0065	0.0045	0.0066	0.0062
Chetan	AFCSS	0.3902	0.6048	0.5906	0.6161	0.8087	0.8063	0.8734	0.9158	0.9107
	MAE	0.0017	0.0026	0.0025	0.0026	0.0033	0.0033	0.0033	0.004	0.0039
Saylee	AFCSS	0.3234	0.4629	0.4579	0.4296	0.5421	0.5429	0.4182	0.5416	0.5679
	MAE	0.0012	0.0020	0.0019	0.0016	0.0023	0.0023	0.0015	0.0026	0.0026
Karishma	AFCSS	0.3577	0.5425	0.5434	0.5540	0.6837	0.6673	0.7600	0.8737	0.7778
	MAE	0.0027	0.0044	0.0043	0.0042	0.0058	0.0055	0.0055	0.0071	0.0068

TABLE II COMPARISON OF LBG, KPE AND FCG ALGORITHM WITH RESPECT TO TOTAL NUMBER OF COMPARISONS, TOTAL NUMBER OF ED COMPUTATIONS AND TOTAL CPU UNITS REQUIRED

Complexity Parameters	FCG	LBG	KPE
Total Comparisons	NM	2MN	2MN
Total No. of ED	0	2MN	2MN
Total CPU units	NM	4MN(10k-1)	4MN(10k-1)

TABLE III TOTAL CPU UNITS REQUIRED FOR ALL SPEECH SAMPLES

Speakers	Codebook Size 256								
	Vector Dimension 8			Vector Dimension 16			Vector Dimension 32		
	No. of Training Vectors	FCG	LBG / KPE	No. of Training Vectors	FCG	LBG / KPE	No. of Training Vectors	FCG	LBG / KPE
HBK	6179	1581824	499856384	3089	790784	502938624	1544	395264	504356864
Tanuja	4940	1264640	399626240	2470	632320	402155520	1235	316160	403420160
Sudeep	3200	819200	258867200	1600	409600	260505600	800	204800	261324800
Chetan	5840	1495040	472432640	2920	747520	475422720	1460	373760	476917760
Saylee	3480	890880	281518080	1740	445440	283299840	870	222720	284190720
Karishma	8820	2257920	713502720	4410	1128960	718018560	2205	564480	720276480

IV. CONCLUSION

In this paper we have used 8 KHZ, 16 bit speech signal which results in very fine quantization levels (i.e 2^{16}). We have considered here three cases with different vector dimensions 8, 16 and 32 for codebook size 256 resulting in compression ratios 16:1, 32:1 and 64:1 respectively. We have used three VQ algorithms LBG, KPE, and FCG for speech coding. Among these three VQ algorithms FCG is faster as compared to LBG and KPE and further FCG algorithm gives 65% less MAE and approximately 20% less AFCSS as compared to LBG and KPE. Regarding complexity of algorithms LBG and KPE are more complex rather than FCG. Among LBG and KPE it is observed that KPE gives less MAE and AFCSS than LBG. The requirement of CPU units for LBG and KPE algorithms is 300 times more than FCG algorithm. At the same time the CPU units required for FCG algorithm reduce two and four times for vector dimension 16 and 32 respectively as compared to vector dimension 8.

References

[1] Jeng-Shyang Pan, Zhe-Ming Lu, and Sheng-He Sun.: 'An Efficient Encoding Algorithm for Vector Quantization Based on Subvector Technique', *IEEE Transactions on image processing*, vol 12 No. 3 March 2003.

[2] R. M. Gray.: 'Vector quantization', *IEEE ASSP Marg*., pp. 4-29, Apr. 1984.

[3] Y. Linde, A. Buzo, and R. M. Gray.: 'An algorithm for vector quantizer design," *IEEE Trans. Commun*.', vol. COM-28, no. 1, pp. 84-95, 1980.

[4] A. Gersho, R.M. Gray.: '*Vector Quantization and Signal Compressio'*, Kluwer Academic Publishers, Boston, MA, 1991.

[5] Chin-Chen Chang, Wen-Chuan Wu, " Fast Planar-Oriented Ripple Search Algorithm for Hyperspace VQ Codebook*", IEEE Transaction on image processing,* vol 16, no. 6, June 2007.

[6] Momotaz Begum, Nurun Nahar, Kaneez Fatimah, M. K. Hasan, and M. A. Rahaman: 'An Efficient Algorithm for Codebook Design in Transform Vector Quantization', *WSCG'2003*, February 3-7, 2003.

[7] Robert Li and Jung Kim: 'Image Compression Using Fast Transformed Vector Quantization', *IEEE Applied Imagery Pattern Recognition Workshop, 2000 Proceedings* 29th Volume , Issue , 2000 Page(s):141 – 145.

[8] Zhibin Pan; Kotani, K.; Ohmi, T., 'Enhanced fast encoding method for vector quantization by finding an optimally-ordered Walsh transform kernel', *ICIP 2005, IEEE International Conference*, Volume 1, Issue, 11-14, Page(s): I - 573-6, Sept. 2005.

[9] Jim Z.C. Lai, Yi-Ching Liaw, and Julie Liu, "A fast VQ codebook generation algorithm using codeword displacement" , *Pattern Recogn.* vol. 41, no. 1, pp 315–319, 2008.

[10] Y.C. Liaw, J.Z.C. Lai, W. Lo, Image restoration of compressed image using classified vector quantization, *Pattern Recogn*. vol. 35, No.(2, pp 181–192, 2002.

[11] N.M. Nasrabadi, Y. Feng, Image compression using address vector quantization, *IEEE Trans. Commun*. vol. 38 No. 12, pp. 2166–2173, 1990.

[12] J. Foster, R.M. Gray, M.O. Dunham, Finite state vector quantization for waveform coding, *IEEE Trans. Inf. Theory* vol. 31, No. 3, pp. 348–359, 1985.

[13] T. Kim, Side match and overlap match vector quantizers for images, *IEEE Trans. Image Process*. vol. 1, No. 2, pp. 170–185, 1992.

[14] J.Z.C. Lai, Y.C. Liaw, W. Lo, Artifact reduction of JPEG coded images using mean-removed classified vector quantization, *Signal Process*. vol. 82, No. 10, pp. 1375–1388, 2002.

[15] K.N. Ngan, H.C. Koh, Predictive classified vector quantization, *IEEE Trans. Image Process.* vol. 1, No. 3, pp. 269–280, 1992.

[16] C.H. Hsieh, J.C. Tsai, Lossless compression of VQ index with search order coding, *IEEE Trans. Image Process*. vol. 5, No. 11, pp. 1579–1582, 1996.

[17] J.Z.C. Lai, J.Y. Yen, Inverse error-diffusion using classified vector quantization, *IEEE Trans. Image Process.* vol. 7, No. 12, pp. 1753–1758, 1998.

[18] P.C. Chang, C.S. Yu, T.H. Lee, "Hybrid LMS-MMSE inverse halftoning technique", *IEEE Trans. Image Process*. vol. 10, No. 1, pp. 95–103, 2001.

[19] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Trans. Multimedia*, vol. 1, no. 3, pp. 264–277, Sep. 1999.

[20] H. Y. M. Liao, D. Y. Chen, C. W. Su, and H. R. Tyan, "Real-time event detection and its applications to surveillance systems," in *Proc. IEEE Int. Symp. Circuits and Systems*, Kos, Greece, pp. 509–512, May 2006.

[21] J. Zheng and M. Hu, "An anomaly intrusion detection system based on vector quantization," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 1, pp. 201–210, Jan. 2006.

[22] Ahmed A. Abdelwahab, Nora S. Muharram, "A Fast Codebook Design Algorithm Based on a Fuzzy Clustering Methodology", *International Journal of Image and Graphics*, vol. 7, no. 2 pp. 291-302, 2007.

[23] M. Satya Sai Ram, P. Siddaiah, and M. Madhavi Latha,(2008, Winter) "Multi Switched Split Vector Quantizer," International Journal of Computer, Information, and Systems Science, and Engineering, WASET, [online],vol. , pp. 1-6, Avaliable: www.waset.org/IJCISSE

[24] Atal. B.S, "The history of linear prediction," IEEE Signal Processing Magazine, Vol 23, pp.154-161, March 2006.

[25] Harma, "A. Linear predictive coding with modified filter structures," IEEE Trans. Speech Audio Process, Vol 9, pp.769-777, Nov 2001.

[26] Viswanathan. R., Makhoul. J, "Quantization properties of transmission parameters in linear predictive systems," IEEE Trans. on Acoustics, Speech Signal Processing,vol 23, no.3, pp. 309-321, June1975.

[27] Stephen. So, & Paliwal. K. K, "Efficient product code vector quantization using switched split vector quantizer," Digital Signal Processing journal, Elsevier, Vol 17, pp.138-171, Jan 2007.

[28] Krishnan. V, Anderson. D.V, Truong. K.K, "Optimal multistage vector quantization of LPC parameters over noisy channels," IEEE Trans. Speech Audio Process, Vol 12, no.1, pp.1-8, Jan 2004.

[29] Biing-Hwang Juang Gray, A., Jr. "Multiple stage vector quantization for speech coding". IEEE Conference on Acoustics, Speech Signal Processing, vol 7, pp. 597-600, May 1982.

[30] H. B. Kekre, Tanuja K. Sarode, "New Fast Improved Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Engineering and Technology, vol.1, No.1, pp. 67-77, September 2008.

[31] H. B. Kekre, Tanuja K. Sarode, "Fast Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Computer Science and Information Technology, Vol. 1, No. 1, pp: 7-12, Jan 2009.

[32] H. B. Kekre, Tanuja K. Sarode, "An Efficient Fast Algorithm to Generate Codebook for Vector Quantization," First International Conference on Emerging Trends in Engineering and Technology, ICETET-2008, held at Raisoni College of Engineering, Nagpur, India, 16-18 July 2008, Avaliable at online IEEE Xplore.

Biographies

Dr. H. B. Kekre has received B.E. (Hons.) in Telecomm. Engineering. from Jabalpur University in 1958, M.Tech (Industrial Electronics) from IIT Bombay in 1960, M.S.Engg. (Electrical Engg.) from University of Ottawa in 1965 and Ph.D. (System Identification) from IIT Bombay in 1970. He has worked Over 35 years as Faculty of Electrical Engg. and then HOD Computer Science and Engg. at IIT Bombay. For last 13 years worked as a Professor in Department of Computer Engg. at TSEC, Mumbai. He is currently Senior Professor working with Mukesh Patel School of Technology, Management and Engineering, NMIMS University, Vile-Parle (W), Mumbai. His areas of interest are Digital Signal processing and Image Processing and Computer Networks. He has more than 200 papers in National / International Conferences / Journals to his credit. Recently five students working under his guidance have received best paper awards.

Ms. Tanuja K. Sarode has Received M.E.(Computer Engineering) degree from Mumbai University in 2004, currently Perusing Ph.D. from Mukesh Patel School of Technology, Management and Engineering, NMIMS University, Vile-Parle (W), Mumbai. She has more than 9 years of experience in teaching. Currently working as a Assistant Professor in Department of Computer Engineering at Thadomal Shahani Engineering College, Mumbai. Her areas of interest are Image Processing, Signal Processing and Computer Graphics. She has 21 papers in National /International Conferences/journal to her credit.