MSVQ: A data compression technique for multimedia applications

Article *in* Journal of scientific and industrial research · December 2006

2 authors:

Singhai Jyoti
Maulana Azad National Institute of Technology, Bhopal

Rakesh Singhai
Rajiv Gandhi Proudyogiki Vishwavidyalaya

Journal of Scientific & Industrial Research
Vol. 65, December 2006, pp. 982-985

MSVQ: A data compression technique for multimedia applications

Jyoti Singhai[1,*] and Rakesh Singhai[2]

[1]Electronics Department, MANIT, Bhopal

[2]Electrical Engineering Department, IIT Delhi, New Delhi 110 016

Received 16 September 2005; revised 13 July 2006; accepted 04 August 2006

Data compression is one of the enabling technologies for the application of communication technology with ever growing Internet, explosive development of mobile communication and the ever-increasing impact of video communication to store the data in a compact way for minimum cost of storage and transmission. There are various compression techniques available which are capable of application specific large compression ratios. This paper presents the Multi Stage Vector Quantization (MSVQ) based on LBG algorithm and Partial Distance Search (PDS) method, which is most suitable for faster encoding required in multimedia applications.

Keywords: LBG algorithm, Multi stage vector quantization, PDS method, Single stage vector quantization

Introduction

In Single Stage Vector Quantization (SSVQ) technique, vector quantization (VQ) is used for data compression, which is achieved by forming vectors from a training data sequence, grouping similar vectors into clusters, and assigning each cluster with a single representative vector. Incoming data is then compressed by replacing vectors with the index of nearest cluster representative. In VQ, clusters are usually referred to as cells and their representatives as code vectors. List of cluster representatives forms a codebook and each index is known as a codeword.

Simple decoder circuit and sufficient amount of compression makes SSVQ an attractive technique for single encoder and multiple decoder scenarios such as in multimedia application, which has many decoders as compared to encoder. Moreover, considerable computational resources are available for the encoding operation, whereas the decoding is to be done in software with the limited amount of computational resource available at the decoder. Hence, to make VQ ideal for multimedia applications, improvement in SSVQ is required to reduce large storage requirement and searching complexity. This is achieved by using Multi Stage Vector Quantization (MSVQ) instead SSVQ. The most prevalent technique for codebook design is the generalized Lloyd algorithm (GLA), which was developed for scalar quantizer design, where each quantizer codeword represented a single sample of the data sequence. But for Lossy compression, encoding sequence of samples or cluster of vectors is more advantageous. GLA was later generalized for use in VQ and is referred as LBG algorithm.

*Author for correspondence
Tel: 91-755-2670480
E-mail: j_singhai@rediffmail.com

Multi Stage Vector Quantization (MSVQ)

SSVQ has an inherent drawback of excessive size of codebook, resulting in an increase in storage complexity. Also, time to search a particular codevector from a large set increases with codebook size. This causes a large coding delay. Alternative to SSVQ is MSVQ or cascade VQ or residual VQ[1]. In MSVQ, quantization process is divided into number of stages. In the first stage, vector quantizer quantizes input vector using a relatively small codebook. The error between input vector and quantized vector is found out and this error is again quantized in the second stage of VQ using a different codebook. Next stage quantizes the error between input vector of the previous stage and quantized output of the previous stage. Indices corresponding to the code vectors in each stage are transmitted over the channel. Decoder finds out the code vectors from pre-stored codebook corresponding to the indices it receives and adds them up to reconstruct the quantized approximation of the original input vector.

In MSVQ[2] (Fig. 1) first stage vector quantizer Q_1 quantizes input vector X and quantized approximation of X is denoted by $\hat{X}_1$. Error vector E_2 is found out by

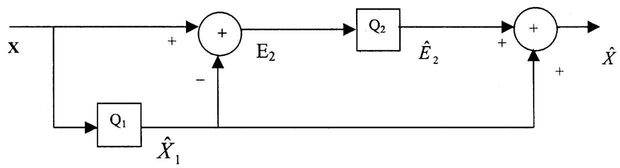

Fig. 1—Block diagram of MSVQ

subtracting output of the first stage vector quantizer $\hat{X}_1$ from input vector X. E_2 is again quantized by the second stage vector quantizer Q_2 yielding a quantized output $\hat{E}_2$. Encoder, instead of transmitting $\hat{X}_1$ and $\hat{E}_2$, simply transmits the indices corresponding to $\hat{X}_1$ and $\hat{E}_2$. Decoder generates $\hat{X}_1$ and $\hat{E}_2$ based on the indices received and finds the overall approximation $\hat{X}$, of input X, by summing $\hat{X}_1$ and $\hat{E}_2$. Commonly used Full Search (FS) or Tree Search (TS) encoding algorithm impose practical limits on codebook size and compression rate due to large computational cost and processing time in searching code vectors $\hat{X}_1$ and $\hat{E}_2$. MSVQ reduces size of codebook, which in turn reduces storage and computational complexity. Partial Distance Search (PDS) method used for encoding reduces processing time. PDS[7] consists of simple modification in calculating nearest neighbor distance. During calculation of the distance sum, if the partial distance exceeds distance to the nearest neighbor found so far, calculation is aborted and processing time is saved.

Code Book Design for MSVQ using LBG Algorithm

In MSVQ, a separate codebook has to be designed for each stage. Codebook for first stage is designed for original training set X using proposed Linde-Buzo-Gray (LBG) algorithm[3]. Next, vectors from original training set X are quantized using codebook of first stage quantizer and a new training set E_2 is formed by computing the difference between original training set X and its quantized value. Training set E_2 is used to design codebook for second stage using the same algorithm. This process is repeated for designing codebooks for successive stages[4].

LBG algorithm, used for designing codebooks for each stage in MSVQ technique, is the generalized algorithm to the case where inputs are not scalar and the distribution is known i.e. training set is available[5]. LBG algorithm or generalized Lloyd algorithm (GLA) is similar to that of k mean algorithm. LBG forms the basic of the most VQ designs. It is defined as:

Step 1

Start with an initial set of reconstruction values $\{Y_i^{(0)}\}_{i=0}^{M}$ and a set of training vectors $\{X_n\}_{in=1}^{N}$. Set k=0, D $^{(0)}$ = 0. This threshold $\in$ will decide the size of codebook.

Step 2

Find the quantization regions $\{V_i^{(k)}\}_{i=1}^{M}$ given by:

$$V_i^{(k)} = \{X_n : d(X_n, Y_j) < d(X_n, Y_j) \forall j \neq i\},\quad i=1,2,\ldots,M.$$

It is assumed that none of the quantization regions are empty.

Step 3

Compute the distortion $D^{(k)}$ between the training vectors and the representative reconstruction values:

$$D^{(k)} = \sum_{i=1}^{M} \int_{V_i^{(k)}} \left\| X - Y_i^{(k)} \right\|^2 f_x(X)\,dX$$

Step 4

If $\dfrac{(D^{(k)} - D^{(k-1)})}{D^{(k)}} <\in$, stop; otherwise continue.

Step 5

For k = k+1, find new reconstruction values $\{Y_i^{(k)}\}_{i=1}^{M}$ that are average value of the elements of each of the quantization regions $\{V_i^{(k-1)}\}$. Go to Step 2.

Performance and Results of the Proposed MSVQ

Performance of a quantizer generally improves when the correlation between the input samples increases. However, performance of MSVQ in the successive stages tends to be less correlated. Quantization error vector is more random in nature than the input vector to the quantizer since error vector components tend to be less statistically dependent. Thus, in practice, MSVQ has only two and occasionally three stages[6]. To simulate and evaluate the performance of proposed technique, a source code is developed in MATLAB for two stage MSVQ and

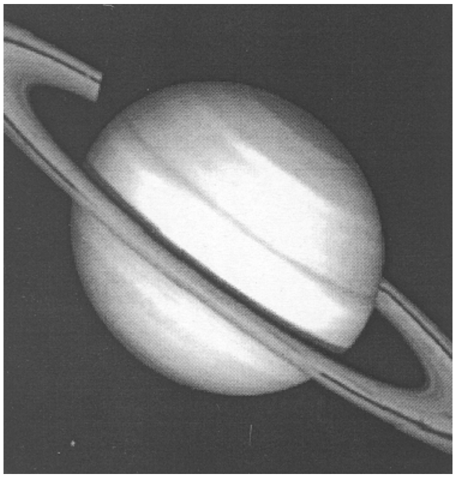

Fig. 2—Reconstructed Saturn image after MSVQ

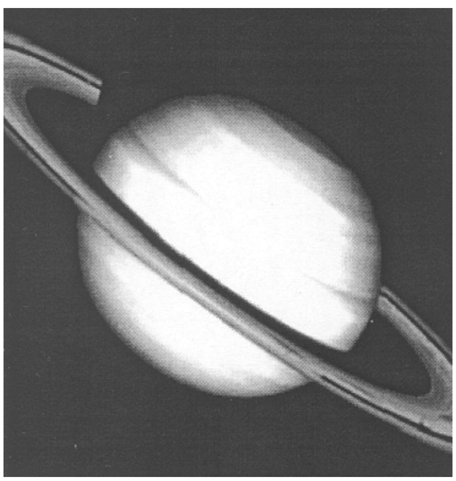

Fig. 3—Reconstructed Saturn image after SSVQ

for single stage VQ technique. The developed code is applied to two most common types of data used in multimedia applications, images and audio signal data.

Using MSVQ in place of SSVQ, a compromise is made between the storage/encoding complexity and the reconstruction accuracy. But for the same size of codebook, image reconstructed using MSVQ (Fig. 2) is better than the image reconstructed using SSVQ (Fig. 3) for the bitmap size of 65536. Bitmap size of the original image was 202536. For the same compression score, gray scale Saturn image compressed using MSVQ is showing more details compared to image compressed using SSVQ. Proposed technique of compressing data using MSVQ was also tested for compressing audio signals. Searching and storage complexity in MSVQ is considerably reduced as compared to that in the SSVQ technique because of reduced codebook size (Table 1). Codebook size for SSVQ at 1.20 bits/sample was 18176, while in MSVQ it was just 3840 for first stage and 2624 for second stage.

Table 1—Comparison of codebook size of audio coders using single stage (SSVQ) and multistage (MSVQ) techniques

S No	Bits per sample	Codebook size MSVQ			SSVQ
		I Stage	II Stage	Stage I + II	
1	1.20	3840	2624	6464	18176
2	1.29	3520	2192	5712	22528
3	1.35	3520	1136	4656	28160

Table 2—Performance of audio coders using single stage (SSVQ) and multistage (MSVQ) techniques

S No	Bits per sample	Distortion in decoded signal, % MSVQ		SSVQ
		After I stage	After II stage	
1	1.20	34.22	17.70	8.30
2	1.29	16.75	10.71	7.86
3	1.35	15.75	8.35	7.50

Performance of the audio coders using the single stage (SSVQ) and multistage (MSVQ) techniques is compared for the percentage distortion in the decoded audio signal for both the SSVQ and the MSVQ techniques at different bit rates (Table 2). Distortion is reduced as number of stages is increased in MSVQ technique. Distortion after second stage in MSVQ is reduced to almost half as compared to distortion after first stage. At lower bit rates (1.20 bits/sample and 1.29 bits/sample), distortion in the MSVQ decoded signal is very large as compared to that in the SSVQ decoded signal. However, at a higher bit rate (1.35 bits/sample), distortion in MSVQ decoded signal is 8.35, which is slightly more than in SSVQ decoded signal (7.5). Original male voice and reconstructed male voice after MSVQ (Fig. 4) and after SSVQ (Fig. 5) are plotted with reference to 71680 samples on x-axis.

Average encoding time is compared for male voice signal (Table 3) and gray scale Saturn image (Table 4) for FS-SSVQ, TS-SSVQ, PDS-SSVQ and PDS-MSVQ, for three bit rates (1.20,1.29 and 1.35 bps) and two vector dimensions (8 and 16 bits). In comparison to FS-SSVQ, TS-SSVQ and PDS-SSVQ, average encoding time is found much reduced in PDS-MSVQ.

Table 3—Average encoding time for male voice signal

S No	Algorithm	8 bits dimension, m sec			16 bits dimension, m sec		
		1.20 bps	1.29 bps	1.35 bps	1.20 bps	1.29 bps	1.35 bps
1	FS-SSVQ	2.954	4.783	7.279	5.684	9.278	14.544
2	TS-SSVQ	2.132	3.432	5.521	4.092	6.587	10.180
3	PDS-SSVQ	1.872	3.021	4.904	3.637	5.752	9.017
4	PDS-MSVQ	1.101	1.737	2.805	1.405	2.197	3.408
Decrease in encoding time in PDS-SSVQ and PDS-MSVQ, %		41.2	42.5	42.8	61.4	61.8	62.2

Table 4—Average encoding time for Saturn image

S No	Algorithm	8 bits dimension, m sec			16 bits dimension, m sec		
		1.20 bps	1.29 bps	1.35 bps	1.20 bps	1.29 bps	1.35 bps
1	FS-SSVQ	2.085	3.134	5.253	4.709	8.365	13.570
2	TS-SSVQ	1.501	2.225	3.677	3.390	5.955	9.553
3	PDS-SSVQ	1.344	1.977	3.262	3.023	5.278	8.454
4	PDS-MSVQ	0.825	1.201	1.976	1.406	2.412	3.942
Decrease in encoding time in PDS-SSVQ and PDS-MSVQ, %		38.6	39.3	39.4	53.5	54.3	54.5

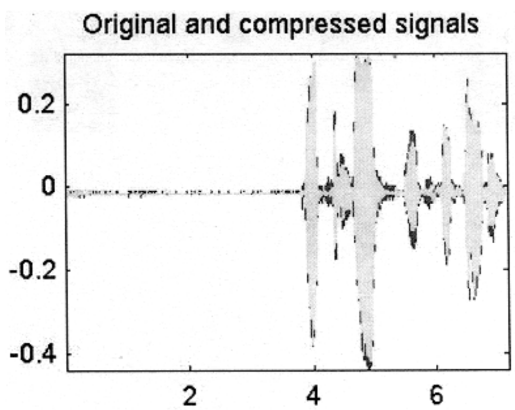

Fig. 4—Original male voice and reconstructed male voice after MSVQ

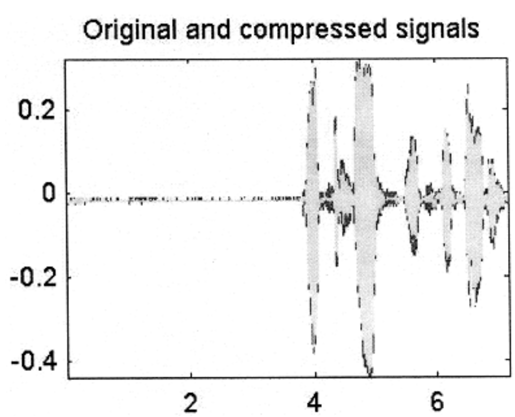

Fig. 5—Original male voice and reconstructed male voice after SSVQ Conclusions

Conclusions

Performance of the proposed technique using MSVQ in terms of percentage distortion, storage requirement and processing complexity is comparable with SSVQ technique for same compression score. The percentage distortion in decoded signal after second stage of MSVQ for higher bit rate is only slightly higher as compared to percentage distortion for same bit rate in SSVQ. Storage requirement in MSVQ is reduced due to reduced size of codebook and encoding time is reduced due to PDS method. This in turn reduced the processing complexity and search time required for decoding for same compression score. This makes MSVQ an attractive alternative for multimedia applications with reduced storage and processing complexity.

References

1 Abut H, *Vector Quantization* (IEEE Press, New York) 1990, 260-300.

2 Veperk P & Bradley A B, An improved algorithm for quantizer design, *IEEE Signal Process Lett*, **7** (2000) 250-252.

3 Linde Y, Buzo A & Gray R, An algorithm for vector quantization design, *IEEE Trans Commun*, **28** (1980) 84-95.

4 Wu X & Guan L, Acceleration of the LBG algorithm, *IEEE Trans Commun*, **42** (1994), 1518-1523.

5 Touzani A, Tong L, Casas R A & Johnson C R, Vector-CM stable equilibrium analysis, *IEEE Trans Signal Processing Lett*, **7** (2000) 31-33.

6 Gray R M, Vector quantization, *IEEE Acoustics, Speech & Signal Process Mag*, **1** (1984) 4-29.

7 McNames J, Rotated partial distance search for faster VQ encoding, *IEEE Signal Process Lett*, **7** (2000) 244-246.