Vector Quantization Using Reflections of Triangular Subcodevectors

Vishnu Makkapati
Honeywell Technology Solutions Lab
151/1, Doraisanipalya, Bannerghatta Road
Bangalore - 560 076, India
vishnu.makkapati@honeywell.com

Pravas Mahapatra
Department of Aerospace Engineering
Indian Institute of Science
Bangalore - 560 012, India
pravas@aero.iisc.ernet.in

Abstract

The design of a codebook consisting of codevectors is the goal of Vector Quantization (VQ) schemes. This solution is not unique and many variants of VQ have been proposed. VQ suffers from computational and memory complexities that increase with the size of codebook and codevector dimensions. The lack of a universal codebook that works across different class of images necessitates the transmission of the codebook along with the codevector indices. This paper proposes a novel method of vector quantization using reflections of triangular subcodevectors. This method jointly reduces the memory and computational resources required for VQ. Numerical results are presented for the proposed scheme in comparison to a traditional VQ method and a method based on reflections.

1. Introduction

Several lossy and lossless schemes have been proposed for image coding. These can be classified as transform based and non-transform based schemes. Discrete Cosine Transform, Karhunen Loeve Transform and Discrete Wavelet Transform are the popularly used transform based schemes. These transforms decorrelate an image and the coefficients thus obtained are encoded to realize coding gains.

Among the non-transform methods, Vector Quantization (VQ) and Scalar Quantization (SQ) are widely used [4]. VQ encodes a sequence of samples while SQ encodes a sample. VQ exploits both linear and non-linear dependencies in an image and is preferred over SQ. VQ and SQ are commonly used to encode the coefficients obtained after applying a transform.

VQ is based on the principle of block coding. The goal of VQ is to represent a set of image vectors using a representative set of vectors known as *codevectors*. The design of a *codebook* is a very challenging problem. VQ schemes assume the availability of a codebook at both the receiving and transmitting ends. Only the indices of the codevectors are transmitted and hence VQ reduces the amount of data required to represent an image.

There are many situations in which a codebook may have to be transmitted along with the codevector indices. In adaptive VQ the codebook is periodically communicated to the receiving end [5]. An universal codebook that works well across all classes of images does not exist [8]. This requires the generation of a codebook for an image or a class of images and transmission of it along with the indices. Hence schemes have been proposed to reduce the size of a codebook [2].

The computational and memory requirements of a VQ scheme increase with the size of codebook and the dimensions of codevectors. VQ has been outperformed by several sub-optimal methods that minimize either of these two resources. Hence, many variants of VQ have been designed to reduce these costs [3, 1, 7].

This paper proposes a method to jointly reduce the size of a codebook and computational complexity. This scheme first divides a codevector into triangular subcodevectors along the diagonal axes. The reflections of these subcodevectors about the diagonal axes are utilized to generate orientations of a codevector.

2. Proposed Scheme

The reflections of a vector are valid vectors in most of the situations. This is true in the case of image vectors also. The vectors of an image exhibit redundancy between their subvectors. The proposed scheme exploits this redundancy to reduce the size of a codebook by using reflections of subvectors.

Let $\mathbf{M}$ be a square vector of dimension $n \times n$ specified as

0-7695-2521-0/06/$20.00 (c) 2006 IEEE

$$\mathbf{M} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{bmatrix} \quad (1)$$

The proposed scheme divides **M** into four triangular subvectors **A**, **B**, **C**, and **D** along the diagonal axes as

$$\mathbf{M} = \begin{bmatrix} & \mathbf{A} & \\ \mathbf{D} & & \mathbf{B} \\ & \mathbf{C} & \end{bmatrix} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{bmatrix} \quad (2)$$

The reflections of these subvectors about the diagonal axes are used to derive orientations of vector **M**. Orientations are generated by combining these reflections in *clockwise* or *anti-clockwise* directions. The reflections of subvectors taken in any of these directions when arranged in a cyclic order will result in four orientations.

Reflection of a subvector produces its mirror image. Every reflection has a mirror line and in this case a diagonal axis acts as a mirror line. The elements along a diagonal axis belong to both the subvectors adjacent to it. In this paper these elements are reflected such that the diagonal axis is retained in one of the adjacent subvectors.

The elements of only one of the diagonal axis are included in a subvector while generating its reflection and the elements of the other diagonal axis are included in the subvector adjacent to it in a given direction (clock-wise or anti-clockwise). These orientations when applied to a vector four times would generate the original vector and form a cyclic group of order four.

This scheme when applied to a vector of dimension 4×4 results in four orientations as

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \quad \begin{bmatrix} a_{4,4} & a_{2,1} & a_{3,1} & a_{4,1} \\ a_{4,3} & a_{3,3} & a_{3,2} & a_{1,3} \\ a_{4,2} & a_{2,3} & a_{2,2} & a_{1,2} \\ a_{1,4} & a_{2,4} & a_{3,4} & a_{1,1} \end{bmatrix}$$

$$\begin{bmatrix} a_{1,1} & a_{4,3} & a_{4,2} & a_{1,4} \\ a_{3,4} & a_{2,2} & a_{2,3} & a_{3,1} \\ a_{2,4} & a_{3,2} & a_{3,3} & a_{2,1} \\ a_{4,1} & a_{1,3} & a_{1,2} & a_{4,4} \end{bmatrix} \quad \begin{bmatrix} a_{4,4} & a_{3,4} & a_{2,4} & a_{4,1} \\ a_{1,2} & a_{3,3} & a_{3,2} & a_{4,2} \\ a_{1,3} & a_{2,3} & a_{2,2} & a_{4,3} \\ a_{1,4} & a_{3,1} & a_{2,1} & a_{1,1} \end{bmatrix} \quad (3)$$

The scheme derives additional orientations by swapping reflections of two adjacent subvectors while retaining the other two subvectors unchanged. The two orientations thus generated when reflections about the first diagonal axis are considered are

$$\begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} & a_{1,4} \\ a_{1,2} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{1,3} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \quad \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{4,2} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{4,3} \\ a_{4,1} & a_{2,4} & a_{3,4} & a_{4,4} \end{bmatrix} \quad (4)$$

The swapping of reflections about the second diagonal axis results in two orientations as

$$\begin{bmatrix} a_{1,1} & a_{3,4} & a_{2,4} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{1,3} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{1,2} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \quad \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{4,3} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{4,2} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{3,1} & a_{2,1} & a_{4,4} \end{bmatrix} \quad (5)$$

Two of the subvectors that are above and below a diagonal axis when combined together form upper and lower triangular matrices respectively. Orientations can be obtained by swapping the reflections of upper and lower triangular matrices about the diagonal axes as

$$\begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} & a_{4,1} \\ a_{1,2} & a_{2,2} & a_{3,2} & a_{4,2} \\ a_{1,3} & a_{2,3} & a_{3,3} & a_{4,3} \\ a_{1,4} & a_{2,4} & a_{3,4} & a_{4,4} \end{bmatrix} \quad \begin{bmatrix} a_{4,4} & a_{3,4} & a_{2,4} & a_{1,4} \\ a_{4,3} & a_{3,3} & a_{2,3} & a_{1,3} \\ a_{4,2} & a_{3,2} & a_{2,2} & a_{1,2} \\ a_{4,1} & a_{3,1} & a_{2,1} & a_{1,1} \end{bmatrix} \quad (6)$$

Additional orientations may be obtained by considering the reflections of these subvectors about horizontal and vertical axes.

It should be noted here that the subvectors in equations (4) and (5) are the same as those in equation (3). Hence the deviations of an input image vector with these orientations can be calculated with out any additional computational effort. Hence the computational complexity does not increase with the size of the codebook when more than four orientations are considered.

It may be argued that any ordering of the elements in a vector would result in a valid vector. This is not true for image vectors as a random ordering of the elements would destroy the inherent structure of a vector. The proposed scheme imposes some structure in the codebook by triangular symmetrization.

The orientations in equations (3), (4), (5), and (6) can be stored in a look-up-table (LUT). An index to the LUT when transmitted along with a vector designates a specific orientation.

The basic procedure for the design of codebook can be stated as:

Encoding procedure

Given an input vector $\mathbf{x}$, the codebook $\mathbf{C} = \{ \mathbf{c}_1, ..., \mathbf{c}_k \}$, and the orientations $\mathbf{T} = \{ T_1, ..., T_r \}$

1. find the indices of optimal orientation (u_{opt}) and codevector (i_{opt})
   $\{u_{opt}, i_{opt}\} = \arg\min_{u=1,...,r;\, i=1,...,k} \{ d(\mathbf{x}, T_u(\mathbf{c}_i)) \}$

Decoding procedure

Given the indices u_{opt} and i_{opt}, the codebook $\mathbf{C} = \{ \mathbf{c}_1, ..., \mathbf{c}_k \}$, and the orientations $\mathbf{T} = \{ T_1, ..., T_r \}$

1. reproduce the input vector as $\hat{\mathbf{x}} = T_{u_{opt}}\left(\mathbf{c}_{i_{opt}}\right)$

Figure 1. Encoding and decoding procedure

Given an input image $\mathbf{X} = \{ \mathbf{x}_1, ..., \mathbf{x}_N \}$ where $\mathbf{x}_i$, $i = 1, ..., N$ are of size $n \times n$

1. Select r, the number of orientations and the orientation operators $\mathbf{T} = \{ T_1, ..., T_r \}$.
2. Select k, the number of clusters and a distortion threshold $\epsilon \geq 0$. Set $m = 0$ and $D_{-1} = \infty$
3. Select k initial cluster centers $\mathbf{C} = \{ \mathbf{c}_1, ..., \mathbf{c}_k \}$
4. Determine minimum distortion partition $P(\mathbf{C}) = \{S_i; i = 1, ..., k\} : T_u(\mathbf{x}) \in S_i$ if $d(\mathbf{x}, T_u(\mathbf{c}_i)) \leq d(\mathbf{x}, T_v(\mathbf{c}_j))$ for all j, u, and v.
5. Calculate the average distortion, $D_m = D(\{\mathbf{C}, P(\mathbf{C})\})$
6. if $\frac{D_{m-1} - D_m}{D_m} \leq \epsilon$, stop with $\mathbf{C}$ and $P(\mathbf{C})$ specifying the final quantizer. Else continue.
7. Set $\mathbf{c}_i = \sum_{i=1}^{p_i} \frac{T_u(\mathbf{x})}{p_i}$, for all $i : T_u(\mathbf{x}) \in S_i$ and p_i is the number of elements in S_i. Set $m = m + 1$ and go to step 3.

The encoding and decoding procedures for the scheme are summarized in Fig. 1.

The orientations in equations (3) to (6) can be further applied on one another to generate additional orientations. Hence this scheme can be utilized to reduce the size of a codebook to any desired level depending on the choice of orientations. Further, the orientations in equations (3) to (6) are not commutative. If T_u and T_v denote orientations u and v respectively, then $T_u(T_v(\mathbf{M})) \neq T_v(T_u(\mathbf{M}))$.

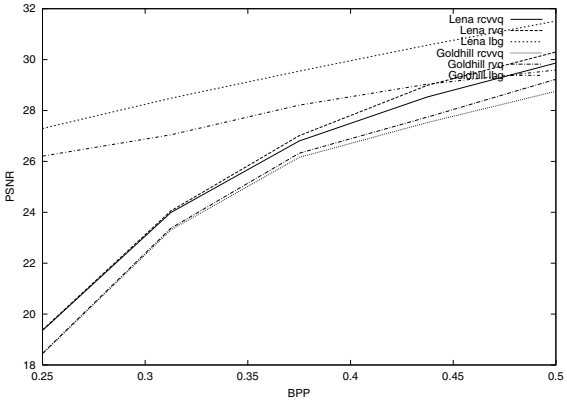

Figure 2. PSNR vs. BPP for RCVVQ, RVQ and LBG schemes

3. Numerical Results

The proposed scheme (RCVVQ) has been tested using many standard test images. The eight orientations listed in equations (3) to (5) have been used in the experiments. The performance of the proposed scheme is compared to that of Linde, Buzo and Gray (LBG) scheme [6] and Reflected VQ (RVQ) [2]. The performance has been quantitatively and qualitatively assessed using Peak Signal to Noise Ratio (PSNR) and visual comparison of original and reconstructed images respectively. The codebooks for these schemes have been initialized with vectors randomly taken from an input image.

Let $M \times N$ and $m \times n$ be the dimensions of an image to be vector quantized and the codevector respectively and B denote the Bits Per Pixel(BPP). If k and r are the number of the codevectors and orientations respectively used in the proposed scheme, a traditional VQ scheme (such as LBG) can use a codebook consisting of $k \times r$ codevectors at the same bit rate.

The ratio of the memory required to transmit the codebook to the total memory required to transmit the codebook and codevector indices defined as *Codebook Storage Ratio* (CSR) is used as a metric to quantify the saving achieved. The CSRs for LBG and proposed schemes are given in equations(7) and (8) respectively.

$$CSR_{LBG} = \frac{k \times r \times m \times n \times B}{\frac{M}{m} \times \frac{N}{n} \times \log_2(k \times r) + k \times r \times m \times n \times B} \quad (7)$$

$$CSR_{proposed} = \frac{k \times m \times n \times B}{\frac{M}{m} \times \frac{N}{n} \times \log_2(k \times r) + k \times m \times n \times B} \quad (8)$$

PSNR vs. BPP plots in Fig. 2 suggest that the proposed scheme offers quality equal to that of RVQ. The PSNR for both these schemes is comparable to that of LBG and it fast

(a) Original Image (b) RCVVQ Scheme (c) RVQ Scheme (d) LBG Scheme

Figure 4. Comparison of original and reconstructed images at 0.5 BPP using codevectors of dimension 4×4

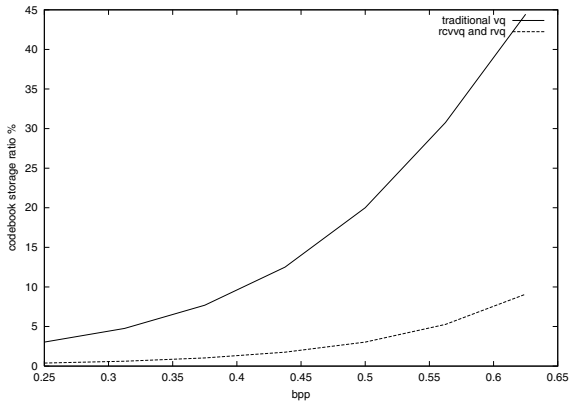

Figure 3. CSR vs. BPP for RCVVQ, RVQ and LBG schemes

approaches that of LBG with an increase in BPP. However, a comparison of CSR for these schemes in Fig. 3 proves that it increases at an exponential rate for LBG while it increases at a much slower rate for the proposed and RVQ schemes. Hence it can be concluded that the scheme offers significant saving in memory with out substantially degrading the quality of an image.

Further, the computational complexity of the proposed scheme is half of that of LBG and RVQ schemes as the deviation of an input image vector with orientations (4) and (5) can be computed by using the deviations of the respective subvectors with those in (3). This reduction in complexity becomes very significant when the size of a codebook and dimensions of codevectors are large. A comparison of the reconstructed images in Fig. 4 shows that the proposed scheme performs equally well as LBG and RVQ schemes.

4. Conclusions

A novel method of VQ using reflections of triangular subcodevectors has been proposed. The proposed method reduces the memory required for a codebook significantly by generating orientations of codevectors. The computational complexity of the proposed scheme is half of that of traditional VQ schemes. Results presented for various data sets suggest that the quality of the proposed scheme equals that of RVQ and approaches that of LBG with an increase in BPP.

References

[1] R. Arvind and A. Gersho. Image compression based on vector quantization with finite memory. *Optical Engineering*, 26:570–580, July 1987.

[2] R. Baker. *Vector Quantization of Digital Images*. PhD thesis, Department of Electrical Engineering, Stanford University, 1984.

[3] W. Chan and A. Gersho. Generalized product code vector quantization: A family of efficient techniques for signal compression. *Digital Signal Processing*, 4:95–126, April 1994.

[4] A. Gersho and R. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Press, Norwood, MA, 1992.

[5] M. Goldberg, P. Boucher, and S. Shlien. Image compression using adaptive vector quantization. *IEEE Transactions on Communications*, COMM-34:180–187, February 1986.

[6] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28:84–95, January 1980.

[7] D. Lyons, D. Neuhoff, and D. Hui. Reduced storage tree-structured vector quantization. In *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 602–605, April 1993.

[8] S. Ramakrishnan, K. Rose, and A. Gersho. Constrained-storage vector quantization with a universal codebook. *IEEE Transactions on Image Processing*, 7(6):785–793, June 1998.