

IMAGE COMPRESSION METHODS WITH DISTORTION CONTROLLED CAPABILITIES¹

Tassos Markas²

*Department of Electrical Engineering, Duke University,
Durham, NC 27706*

John Reif

*Department of Computer Science,
Duke University,
Durham, NC 27706*

Abstract

Traditionally, lossy compression schemes have focused on compressing data at fixed bit rates in order to communicate information over limited bandwidth communication channels, or to store information in a fixed-size storage media. In this paper we present a class of lossy data compression algorithms that are capable of encoding images so that the loss of information complies with certain distortion requirements. The developed algorithms are based on the Tree-Structured Vector Quantizers (TSVQ). The first distortion controlled algorithm uses variable-size image blocks, encoded on quad-tree data structures, to efficiently encode image areas with different information content. Another class of distortion controlled algorithms that are presented is based on recursive quantization of error image blocks that represent the difference between the current approximation and the original block. We will also describe the progressive compression properties of these algorithms. Finally, we will present their compression/distortion performance using satellite images provided by NASA, and we will show that they achieve better performance than the TSVQ algorithms at high bit rates.

1 Introduction

Vector Quantization is a block coding technique that is used extensively to compress data at low bit rates. Vector quantizers compress image data by mapping the values of an image block, represented by a discrete vector, into a code that indicates the location of the best-match entry found in a reconstruction codebook[1]. The N-size codebook vectors divide the entire search space into a fixed number of N-dimensional regions that is determined by the size of the vocabulary. The encoding process is responsible for identifying the corresponding region of each input block and replacing the block with the index of the vector that represents that region. The full search method is a computationally expensive task since each vector has to be compared against all codebook entries using a given distortion measure. A more efficient approach that reduces the computational requirements from linear to logarithmic time is the TSVQ algorithm[2]. The reconstruction vocabulary is stored in a binary tree form, where the leaf nodes of the tree represent the codebook entries and the intermediate nodes constitute the centroids of their children nodes. Each codebook entry is represented by a unique index which is the binary path from the root node of the tree to the

¹This work has been supported by the Center of Excellence in Space Data and Information Sciences under NASA/CESDIS NAS 5-30428 URSA Contract. Additional support has been provided by DARPA/ISTO Contract N00014-88-K-0458, and Air Force Contract AFOSR-87-0386.

²Also with Center of Systems Engineering, Research Triangle Institute, RTP, NC 27709

corresponding leaf node. This structure allows images to be compressed at various rates by using the k most significant bits of the index of the best-match vector. This capability is often referred to as “progressive transmission” since the coded stream can be transmitted in a progressive fashion to communicate a distorted version of the original image using low rates. Additional information can be transmitted at a later time to produce a more accurate representation of the image. This property allows the TSVQ to trade-off between compression rates and distortion. The potential compression rate of a TSVQ system is a function of the vector and the vocabulary sizes. The lowest possible distortion is achieved at $\frac{b}{N}$ bpp, while the lowest rate is $\frac{1}{N}$ bpp. This rate creates a highly distorted image since only the most significant bit of the best-match index is used to approximate the image block. In TSVQ the possible compression rates can vary only $\frac{b-1}{N}$ bpp, whereas the quality of the reconstructed image is upper bounded by the distortion achieved at the $\frac{b}{N}$ bpp rate.

For certain applications, where only minimal loss of information can be tolerated, the quality of the decoded image achieved using the highest rate is not adequate. Such applications include target recognition, object measurement, and detailed examination of satellite images. To eliminate the upper bound, posed by the TSVQ in terms of image quality, we have developed a class of distortion-controlled methods that are capable of achieving minimal distortion at high bit rates using a single set of codebooks. The compression rates of the distortion-controlled vector quantizers (DCVQ) vary from low rates, suitable for quick browsing of large amounts of image data, to high rates for reconstructing accurately images with high information content. The first technique that will be presented is the Multi-Resolution Vector Quantization (MRVQ) algorithm that compresses data by dividing an image block into a number of variable-size subblocks, and encoding them separately using a quad-tree representation. Two similar approaches have been proposed in [3, 4]. Our approach uses a variable-size and variable-depth quad-tree representation to utilize large size blocks without reducing the distortion/compression performance at high bit rates. The MRVQ method requires multiple vocabularies that will be used for compressing image blocks at different levels of the quad-tree. The training of these vocabularies is a computationally expensive task, and for that reason we will present methods for creating these product codebooks from a single vocabulary on-the-fly. The second class of DCVQ algorithms that will be presented is based on recursive quantization of error vectors, defined as the difference between the current approximations and the original vectors. We will refer to these algorithms as Error Coding Vector Quantizers (ECVQ), and Error Coding Multi-Resolution (ECMR) methods. The later combines the advantages of both MRVQ, and ECMQ schemes to maximize the compression/distortion performance.

These methods can be incorporated in any data compression system to enhance its compression performance. A typical model of a data compression system is shown in Figure 1. Block transform methods, such as the Discrete Cosine Transform [5], and the Predictive Coding [6, 7] can be used to transform the incoming image to a different domain of representation so that a large fraction of its information is packed in relatively few coefficients. It has also been shown that pruning methods[8, 9] experience better performance than the basic TSVQ scheme. These schemes can also be incorporated in the design of the DCVQ methods to take full advantage of the state-of-the-art data compression algorithms. Finally, a lossless compressor can be used to increase the compression ratios at low bit rates by maximizing the source entropy of the quad-tree codes.

The widely used Mean Squared Error (MSE) measure will be used throughout

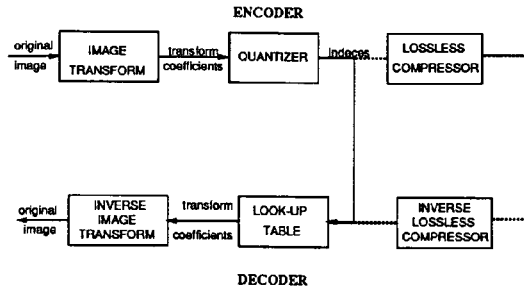


Figure 1: A typical data compression system

this paper to quantify the distortion of the reconstructed images. The Peak Signal-to-Noise Ratio (PSNR) measure will be used to evaluate the performance of the developed algorithms.

$$PSNR = 10 * \log_{10} \left(\frac{255^2}{\sum_{i=1}^{LL} (x_i - \hat{x}_i)^2} \right)$$

where LL is the image size, and x_i , \hat{x}_i are the pixel values of the original and the reconstructed images respectively.

The performance of the developed algorithms was evaluated using images obtained from the USC³ database, as well as Thematic Mapper data obtained from NASA. The required vocabularies were created using the tree-structured version of the k-means algorithm[10].

2 Multi-Resolution Vector Quantization

The classic TSVQ algorithm uses a fixed number of bits to encode all image blocks independently of their information content. However, the amount of information contained in typical images is not uniformly distributed among the image regions. The MRVQ algorithm takes advantage of this property by utilizing large size blocks to compress image areas with low information content at low bit rates, and small size blocks to compress areas with high information content at high bit rates.

During MRVQ, the image is divided into large size blocks ($N \times N$) which are quantized by applying TSVQ in a vocabulary (V_N) that contains $N \times N$ -size vectors. The best-match vector obtained from this operation represents the current approximation of the original vector. If this approximation satisfies the specified distortion criteria, the index of the best-match vector replaces the $N \times N$ -size vector, otherwise the image block is divided into four equal size subblocks (quadrants) which are quantized separately using vectors from a $(N/2) \times (N/2)$ vector vocabulary (V_{N-1}). The same procedure is repeated recursively until the distortion requirements are satisfied in all subblocks. The highest possible resolution is a single pixel (image area is left uncompressed). A V_1 vocabulary containing all the possible intensity values is assumed

³University of Southern California

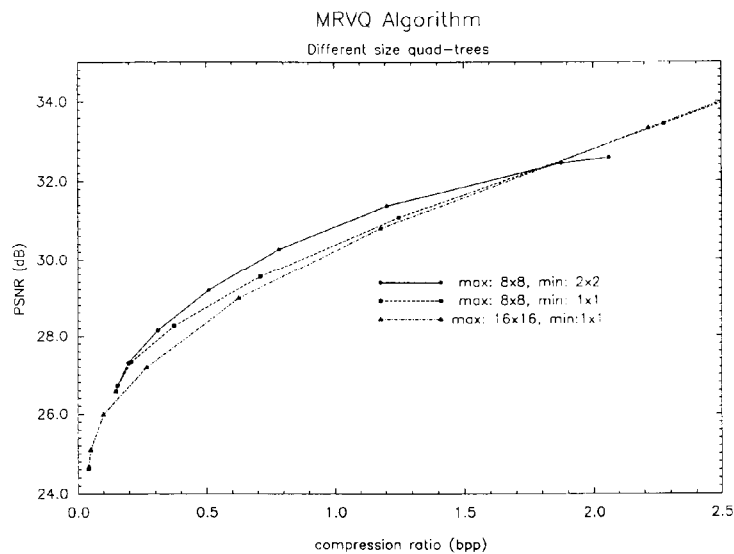


Figure 2:

in this case. The advantage of this recursive decomposition is that by reducing the vector size, the search space is reduced by several orders of magnitude; this results in a more efficient reconstruction of the original image block.

The MRVQ algorithm encodes the indices of the best-match vectors using a quad-tree representation [11]. The quad-tree product VQ method, presented in [3, 4] requires a fixed size tree which is used throughout the compressing process. Our approach differs in that it uses variable size quad-trees to reduce the encoding overhead of the quad-trees in high-detailed areas. This way, large size blocks can be utilized to compress blocks at low rates without affecting the compression performance at high bit rates. With this feature, the MRVQ algorithm can compress image areas with minimal information (i.e., black background of a picture, sea or clouds in an aerial photo) at very low bit rates. The effect of the maximum size image block in terms of compression/distortion performance is depicted in Figure 2. This figure shows the ability of the MRVQ algorithm to achieve low compression rates by using large size blocks without penalizing the compression performance at high rates.

Figure 3 shows the quad-tree representation of two image blocks with different information contents. It can be seen that during compression at high rates only a portion quad-tree needs to be encoded (nodes connected with dashed lines can be eliminated). It should be also noted that a minimum overhead (3 bits per block) is required to indicate the size of the quad-tree structure.

The described top-down MRVQ approach is a rather inefficient method for compressing high-detail areas since the algorithm spends a significant amount of time calculating distortions and identifying best-matches for large size blocks that will not be used in the final bit stream. A more efficient approach is to use a down-up

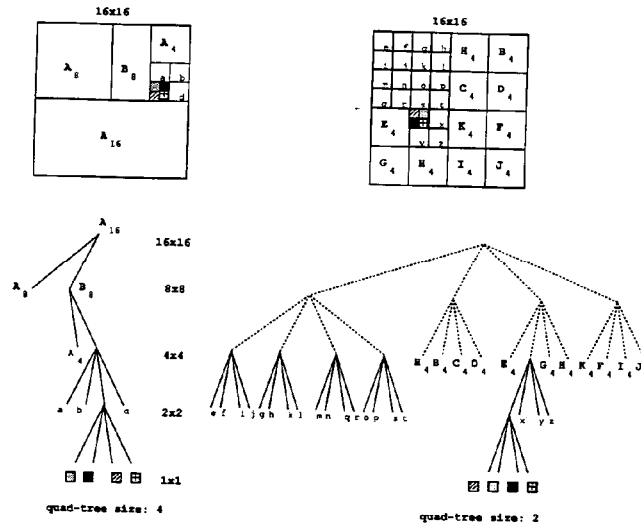


Figure 3: Quad-Tree Representation of the MRVQ Algorithm

approach, where quantization is initially applied at the lowest possible levels of the quad-tree using small size vectors. At each level, the distortion of the best-match vector is examined to verify if it complies with the given criteria. In this case the current best-match vector is used to encode the block, and the current node becomes leaf node of the quad-tree. If the distortion criteria are not met, the best-match vectors of the immediately lower level will be used to encode the quadrants of the current node.

The performance of the MRVQ method can be further improved if additional efficiency measures are applied during the decomposition of an image block in the quad-tree structure. One method is to trade distortion for compression rate by eliminating the subtrees of the quad-tree structure that fail to achieve significant reduction of the block distortion. A similar pruning scheme has been used in [9] to eliminate nodes of the tree-structured codebook. The pruning of the quad-tree structure is accomplished as follows: At each node of the quad-tree, we examine the distortion improvement and compression rate of each subtree. If the ratio of these two quantities is below a predefined threshold, the subtree is deleted and the best-match vector of the current node is used to approximate the examined subblock.

$$\frac{d(MSE)}{d(rate)} < threshold$$

The pruning algorithm improves the compression performance at medium and high bit rates. The performance of the pruning scheme is shown in Figure 4.

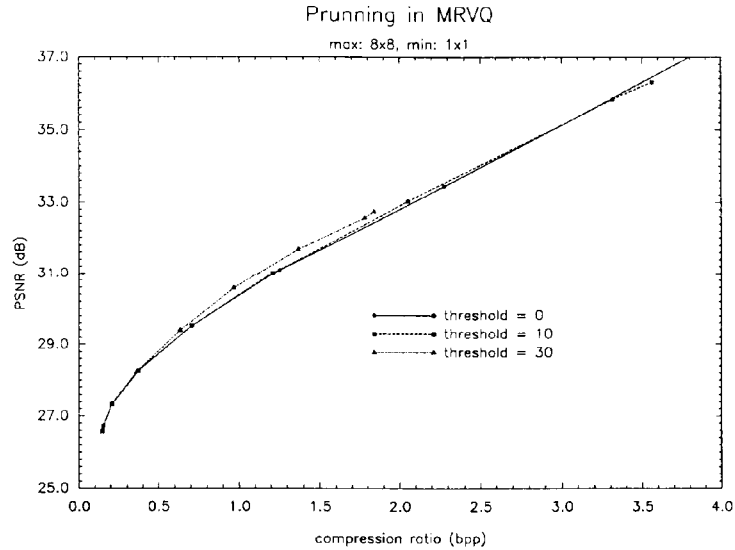


Figure 4:

2.1 Generation of Product Vocabularies

The MRVQ algorithm uses different vocabularies at each level of the quad-tree which are trained with different sets of data. As was mentioned earlier, the training of these vocabularies is a computationally expensive process. An alternative approach is to generate these vocabularies on-the-fly using only a single vocabulary that will be referred as “core” vocabulary.

A number of linear interpolation methods have been evaluated for generating product codebooks from the “core” vocabulary. The first method is the *replication* scheme [12], where a $N \times N$ size block is expanded to a $(2N) \times (2N)$ block by replicating adjacent pixels. Another method that offers smoother transitions among the neighboring pixels is the *bilinear* interpolation [13]. The disadvantage of this scheme is its inaccurate generation of pixels in the right, and lower sides of the image block. This effect is magnified during the generation of large size vector vocabularies. To overcome the edge problem of the bilinear transformation we modified this method so that the left and lower side pixels of each block are left empty. Quantization is applied on the rest of the pixels, and a post-processing averaging method is used to fill these empty areas. The advantage of this method is that it reduces the vector quantization effect (sharp discontinuities among adjacent blocks) for large vectors by blurring the image those areas.

Figure 5 shows the compression/distortion performance of evaluated interpolation methods. The tree-structured version of the k-means algorithm was used to train the 2×2 -size vocabulary. The remaining vocabularies were generated by the 2×2 vocabulary using the described methods. Figure 5 shows that the replication method achieves almost identical performance with the separate vocabulary MRVQ case, and

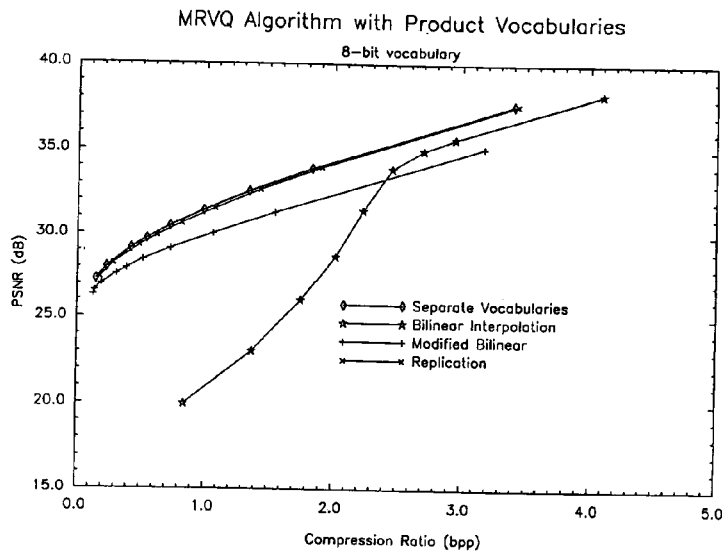


Figure 5:

it is also the simplest one to compute.

3 Error Coding Vector Quantizer (ECVQ)

This scheme is an iterative procedure that encodes the error vectors between the current approximation and the original vector using a TSVQ quantizer. The basic idea behind this scheme is to reduce the block variance at each iteration step so that blocks can be efficiently approximated using a minimum number of iterations. At each step, the current approximation is compared against the original vector to determine if the distortion requirements are met. The recursive formula that describes the ECVQ method is:

$$\hat{x}_i = x_{i-1} + BM(\bar{x} - x_{i-1}), \quad \text{and} \quad \hat{x}_0 = BM(\bar{x})$$

\bar{x} is the input vector, \hat{x}_i is the i th approximation of the input vector, and BM is function that identifies the best-match from the appropriate vocabulary.

The initial vocabulary was created using vectors obtained from the intensity domain of the training set of images. The remaining vocabularies were created using different size vectors obtained from error images to capture the fluctuation of the variance in different iteration levels. The codebook that is used during the n th iteration is created using a set of $N \times N$ -size vectors taken from error images that were generated by compressing the original set with a $(N/n) \times (N/n)$ TSVQ compressor.

The performance of the ECVQ scheme was inferior to the one obtained in the MRVQ case, especially at high bit rates. The primary reason of this behavior is that the vector size remains constant at each iteration step. Because of the constant vector size the ECVQ scheme cannot capture the intensity fluctuations that are present in large size blocks.

4 Error Coding Multi-Resolution Algorithm

A modified version of the previous algorithm is to use the MRVQ method during the encoding of the difference vectors. The advantage of this approach is that the size of the error vectors is reduced each time an additional level of accuracy is introduced in the quad-tree. This way, the variance as well as the search space is reduced at lower levels of the quad-tree. The fact that the variance of the error values is much smaller than the variance of the intensity values implies that the number of lower levels, required to represent efficiently the image block, is significantly reduced. This results in better approximations of the best-match vectors when compression is applied at the lower levels of quad-tree. This algorithm maintains all the important features of the MRVQ method that were described in earlier sections. The error vocabulary, used during the n th level of the quad-tree, was created using a set of $N \times N$ -size vectors taken from error images that were generated by compressing the original set with a $N \times N$ TSVQ compressor.

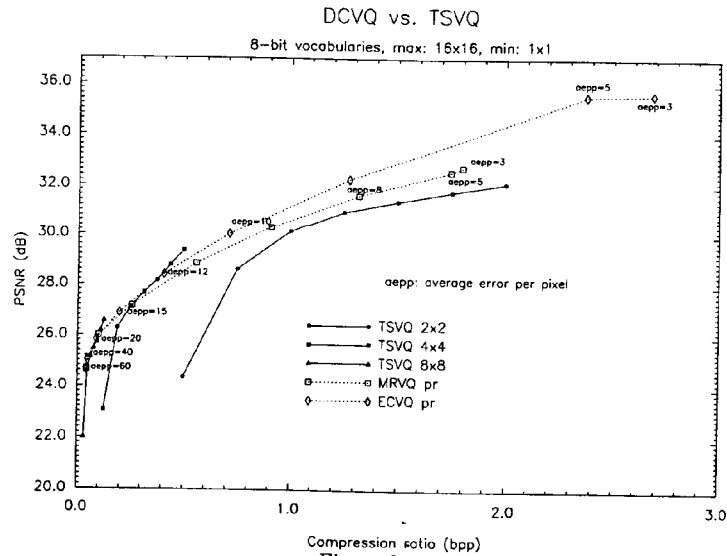
5 Progressive Transmission in DCVQ Schemes

Progressive transmission in the MRVQ, and the ECMR cases can be accomplished with two different approaches. The first approach, similar to the one described in the TSVQ case, is to transmit only the k most-significant bits of the vector indices and leave the quad-tree code intact. The second approach is to transmit partially the quad-tree (one level a time) along with the associated indices of the best-matches. Extremely low rates can be achieved using a combination of both methods.

5.1 Performance Evaluation of the DCVQ Schemes

The required vocabularies were created using the tree-structured version of the k -means algorithm. Each vocabulary had 256 entries and was trained using a set of satellite images containing thematic mapper data provided by NASA. The presented benchmarks were obtained by compressing an image that was not included in the training set.

Figure 6 shows the MRVQ, and the ECMR curves obtained at the lowest possible rates that satisfy the distortion requirements. This figure also shows that both MRVQ, and ECMR exceed the performance of the TSVQ in high rates. Even at low rates both algorithms offer better visual characteristics than the TSVQ scheme by maintaining the sharp edges which are important features in the human visual system. The ECMR method outperforms the MRVQ method at high rates by minimizing the number of small-size vectors that are used to reproduce high-detail areas. This stems from the fact that the number of nodes at a given level of the quad-tree structure grows exponentially with the tree depth (4^n at the n th level). This means that compression at low levels introduce high overhead because of this exponential growth. The ECMR



algorithm manages to compress image blocks at the upper levels of the quad-tree by reducing the variance of the input source and the dimensions of the search space each time a lower compression level is introduced. The DCVQ algorithms were also applied in images obtained from the USC database, and they have experienced similar performance.

6 Conclusions

In this paper we have presented a class of distortion controlled vector quantizers that are capable of compressing images so that they comply with certain distortion requirements. This class of data compression algorithms is applicable in cases where significant loss of information cannot be tolerated. These algorithms can be used in a wide range of applications, from compressing images at low rates for quick browsing of large amounts of data, to high rates for detailed examination of images with high information content. In conclusion, it has been shown that the performance of these algorithms exceeds the performance of classic TSVQ methods at low rates based on the MSE distortion measure.

References

- [1] R.M.Gray, *Vector Quantization*, IEEE ASSP Magazine, Vol.1, Apr. 1984, pp. 4-29.
- [2] R.M.Gray, and H.Abut, *Full Search and Tree Searched Vector Quantization of Speech Waveforms*, Proc. ICASSP, May 1982, pp. 593-596.
- [3] D.J.Vaisey, and A.Gersho, *Variable Blocksize Image Coding*, Proc. ICASSP, Apr. 1987, pp.25.1.1-4.
- [4] C.Chiu, and R.L.Baker, *Quad-tree Product Vector Quantization of Images*, Proc. SPIE, Vol. 1099, Advances in Image Compression and Automatic Target Recognition, Mar. 1989, pp. 142-153.
- [5] N.Ahmed, T.Natarajan, and K.R.Rao, *Discrete Cosine Transform*, IEEE Trans. Comput., Vol. C-23, Jan. 1974, pp. 90-93.
- [6] A.Habibi, *Comparison on n-th order DPCM encoder with linear transformations and block quantization techniques*, IEEE Trans. Commun. Technol., Vol. COM-19, Dec. 1971, pp.948-956.
- [7] H.Hang, and J.W.Woods, *Predictive Vector Quantization of Images*, IEEE Trans. Acous. Speech and Sig. Proc., Vol.33, Nov. 1985, pp. 1208-1219.
- [8] L.Breiman, J.H.Friedman, R.A.Olsen, and C.J.Stone, *Classification and Regression Trees*, The Wadsworth Statistics/Probability Series, Belmont, CA: Wadsworth 1984.
- [9] K.Tzou, *Progressive Image Transmission: A Review and Comparison of Techniques*, Optical Engineering, Vol. 26, July 1987, pp. 581-589.
- [10] Y.Linde, A.Buzo, and R.M.Gray, *An Algorithm for Vector Quantizer Design*, IEEE Trans. on Commun., Vol-28, No.1, Jan. 1980, pp. 84-95.
- [11] A.Klinger, and C.R.Dyer, *Experiments on Picture Representation using Regular Decomposition*, Computer Graphics and Image Processing, Vol. 5, 1976, pp. 68-105.
- [12] A.K.Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989, pp.253.
- [13] J.S.Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall, 1990, pp.495-496.