IMAGE EDGE BLOCK CLASSIFICATION FOR CVQ USING THE SD FILTER

James B. Farison and Mahmoud K. Quweider

Department of Electrical Engineering
The University of Toledo, Toledo, Ohio 43606
USA

ABSTRACT

A novel technique to classify image edge blocks is presented. It is based on defining a set of linearly independent signature vectors with a one to one association with the edge classes. A set of filter vectors emphasizing the projection of one signature vector and suppressing all others is then designed. Classification of an input edge block is accomplished by choosing the index of the filter with the maximum output magnitude. Coded images based on this classification are shown to preserve their quality and enjoy considerable dB gain over two existing methods. The new technique can be easily implemented using a parallel algorithm with little storage requirement.

1. INTRODUCTION

Vector quantization (VQ) [1], [2] is the extension of scalar quantization to vectors. It is mainly used to reduce the transmission bit rate or the storage of waveforms and images while maintaining acceptable quality. Its power stems from the fact that better performance can always be achieved by coding vectors rather than scalars. VQ is simply a mapping Q of a k-dimensional Euclidean space R^k into a finite subset Y of the original space R^k, that is, Q: $R^k \rightarrow Y$ where $Y = \{Y_i : i=1,2,...,N\}$ is the set of reproduction vectors called the codebook and N is the number of vectors in Y. No actual data is sent or stored, rather the index to the entry in the codebook Y that best matches the input vector is sent or stored. Compression is achieved since the index takes fewer bits to represent than the actual data of the vector. The codebook is available at both the transmitting and receiving ends of the transmission or storage system, so when index *i* is received the entry $\mathbf{Y}_i$ is retrieved as the reproduction of the coded vector. Usually the mean square error (MSE) is used as the distortion measure due to its simplicity and analytical tractability. For two vectors **X** and **Y**, the MSE is given by:

$$d(X,Y) = \|X - Y\|^2 = \sum_{j=0}^{k-1} (x_j - y_j)^2. \quad (1)$$

Studies of image coding with VQ have showed two key problems. The first is the high computational complexity which grows exponentially with rate and dimension of the vector, where the number of bits needed to code a pixel is given by Log_2 N/k. The second problem is the edge degradation due to the small number of edge vectors in the codebook. This arises because edges constitute a small fraction of the total image despite their importance in the perception of image quality. Additionally the MSE does not have any edge preserving property or preferential treatment to edges so a codebook with few edge vectors will result in images with poor quality.

Ramamurthi and Gersho [3] introduced the CVQ, which is based on a composite model. It tries to preserve features while retaining a simple distortion measure. The blocks in an image, typically 4x4 or 5x5, are separated into shade and edge blocks. Using a geometric approach, the edge blocks are further divided into various edge classes depending on their orientation, location and polarity (positive for intensity transition from high to low and negative for the other way). A separate codebook C_i for each class is designed from a training set of blocks belonging to that class. Each class may have a different book size N_i such that the total number of codewords in all the classes is equal to N. This CVQ enhances the edge fidelity, reduces the computational complexity and simplifies the codebook design. This is because only a certain subclass of the codebook is searched and each class codebook uses a subset of the whole training set for its design. The classification algorithm is crucial for such systems. In this paper, we present a new method to classify image edge blocks using a special form of the simultaneous diagonalization (SD) filter [4], [5], which is a linear filter known to maximize the projection of the vector form of a selected feature (pattern) while suppressing the projection of the other undesired features (patterns). The design process uses signature vectors that describe the characteristic behavior of the respective features. We first define the set of edge classes (patterns). For each edge class, a signature vector is defined and an SD filter is described. The classification of an input edge block is accomplished by arranging the input block pixel values into a vector, transforming (filtering) the vector, and choosing

0-7803-2431-5/95 $4.00

the filter index among all SD filters which has the maximum output magnitude.

2. SD FILTER-BASED CLASSIFICATION

For the SD filter-based classification, 14 edge classes (corresponding to dark to light transition) are defined as shown in Fig. 1.

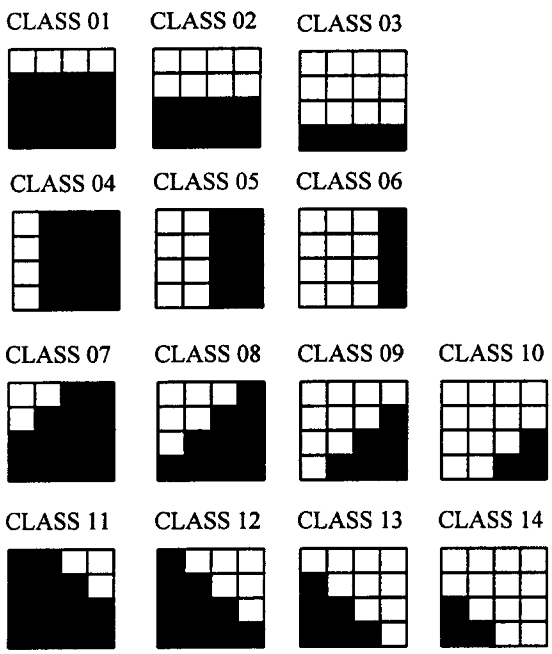


Figure 1. The fourteen edge classes.

There are 3 horizontal, 3 vertical, and 8 diagonal classes. Although the classes are defined here for a 4x4 block size, similar definitions for different block sizes are possible as long as the number of classes is less than or equal to the number of pixels in the block and the resulting signature vectors are linearly independent (necessary conditions for the SD filter design). The signature vector of each class is formed by stacking the rows (in left to right, top to bottom manner) and assigning values of β and $-\beta$ (where β is an arbitrary positive integer) to the bright and dark regions, respectively. The edges shown here correspond to a step transition profile (step-edge model). A ramp transition profile (ramp-edge model), allowing for pixel quantization, blur, and low-pass effects by assigning a one-pixel transition band with value of 0 ($(-\beta+\beta)/2$), can also be used. The 16-dimensional signature vectors from the 4x4 edge class blocks are linearly independent and each corresponds to a specific edge orientation and location.

Using the signature vectors ($\mathbf{v}_1, \ldots, \mathbf{v}_{14}$) for the 14 edge block classes, we form a filter vector, $\mathbf{x}_i$, for each class i [4], [5] :

$$\mathbf{x}_i = \alpha\left[I - U(U^tU)^{-1}U^t\right]\mathbf{d}_i , \qquad (2)$$

where $\mathbf{d}_i=\mathbf{v}_i$ is the specific (desired) class pattern, $U=[\mathbf{v}_j, j\neq i]$ is the 16x13 matrix of the other 13 (undesired) pattern vectors, I is the identity matrix, and α is an arbitrary scalar. This special case (2) of the general SD filter is in the direction of the component of $\mathbf{d}_i$ orthogonal to all of the undesired patterns U ($\mathbf{x}_i^t\mathbf{v}_j=0$, $i\neq j$). With a filter vector designed for each edge block class, that filter gives the largest magnitude when dotted with an actual input edge block vector (after a simple transformation described below) that belongs to that class, while other filters give virtually zero.

To classify each input block, we first decide whether it is a shade or non-shade block. The most common methods to accomplish this use gradient information, mean value, variance or selected transform coefficients of the block. For our classifier, we use a method based on finding the range, R, of the block (maximum gray value - minimum gray value). The method is efficient for our classification since the maximum gray value, g_{max}, and minimum gray value, g_{min}, are also used for the non-shade block transformation to be described shortly. The block range is compared with a pre-selected threshold shade value, T, in the range [15,18] for 8-bit intensity levels. It is classified a shade block if the range is less than T. Otherwise, it is considered a non-shade block and the SD filter-based classification algorithm is applied.

To classify an input non-shade block, we apply the linear mapping

$$g' = 2\beta \ (g - g_{min}) / (g_{max} - g_{min}) - \beta \qquad (3)$$

to transform the gray level values of the block to the range $[-\beta,\beta]$, where g is the original scalar gray level value and g' is the transformed one. In vector notation, each block is represented by

$$\mathbf{g}' = \zeta\mathbf{g} + \kappa\mathbf{1} , \qquad (4)$$

where $\mathbf{g}'$ is the transformed vector, $\mathbf{g}$ is the original vector and $\mathbf{1}$ is a vector of ones. $\zeta = 2\beta / (g_{max} - g_{min})$ and $\kappa = -\beta\ (g_{max} + g_{min}) / (g_{max} - g_{min})$ are block-adaptive scaling and shifting factors, respectively, with κ inversely proportional to the block contrast. This mapping is used to allow the SD filter-based classification to be applied effectively to blocks of different edge values.

The vector, $\mathbf{g}'$, is dotted with each of the SD filter vectors (2), giving $\mathbf{x}_i^t\mathbf{g}'$. A 14 SD-filter bank can accomplish this in parallel, with the input vector available to all of the filters at the same time. The non-shade block is assigned to the class of the filter that gives the maximum absolute output value (this is to include the corresponding bright to dark transition blocks in the same class). If the output from another filter is within percentage ρ (a value determined experimentally) of the maximum value, the block is assigned to a mixed class. Thus, the total number of classes is 16 (1 shade, 1 mixed, 14 edge).

3. SIMULATION RESULTS

The effectiveness of the classification algorithm is demonstrated by implementing an SD filter-based CVQ. The training sets of the classes are created by running the classification algorithm on a training set of vectors from a set of six 512x512 8-bit intensity images of different nature. Experimentally, the value for the shade threshold, T, was set to 16 and percentage ρ to 80. A codebook of size N_i is designed for each class such that the total number, N, of codewords is $\sum_{i=1}^{M} N_i$, where M is the number of classes and N is the fixed-rate codebook size. The exact codebook design procedure, in which the codebooks are built simultaneously in an interconnected manner by growing M unbalanced trees, one for each class, one node at a time, is described in detail in [6]. The node to split among all nodes of the M trees is the one which results in the biggest decrease in average distortion after the split. The splitting process continues until the number of terminal nodes in all of the trees equals the desired codebook size N. We consider the terminal nodes of each tree as the codebook for that class. The codebook size of each class is not known in advance and is determined during the codebook build up.

The method is tested with the Lena image, which is outside the training set. The reconstructed image quality is measured by the peak-signal-to-noise ratio in dB (PSNR), given as:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{V_{ptp}^2}{V_{mse}}\right), \qquad (5)$$

where V_{ptp} is the maximum intensity (255 for 8 bit intensity) and V_{mse} is the mean square error between the original image and the coded image. Results are presented as Method 3 in Table 1, along with previous methods using gradient extraction [3] (Method 1) and DCT coefficients [7] (Method 2). Values shown for Method 1 have been linearly interpolated from [3]. The proposed system provides about 3.5 dB gain over Method 1 and 1.5 dB over Method 2 at the indicated rates. The block transformation needed for our method is computationally less demanding than the gradient extraction and transform coefficients computation of Methods 1 and 2, respectively. The quality of the coded images is preserved. Edges are reproduced faithfully and their jaggedness is greatly reduced. Some blockness is apparent at the low rates but quickly disappears as the rate is increased. The good visual quality of the coded images is attributed to the grouping of perceptually similar blocks by the classifier and the subsequent codebook design of each class. Fig. 2. shows the original Lena image, the reconstructed image at 0.813 bpp and the difference image magnified by a factor of 4 and shifted by 128.

TABLE 1
CVQ SIMULATION RESULTS FOR LENA IMAGE (dB)

Rate (bpp)	Method 1 Gradient Extraction	Method 2 DCT Coefficients	Method 3 SD filter Step-edge model	Method 3 SD-filter Ramp-edge model
0.625	29.35	31.26	32.60	32.58
0.688	29.72	31.79	33.27	33.20
0.750	30.01	32.23	33.76	33.73
0.813	30.45	32.57	34.16	34.10

4. CONCLUSION

We presented a new edge block classification method based on a novel adaptation of the SD filter. Excellent results were obtained by coding images using SD filter-based CVQ as seen by subjective and objective measures. Comparison with two existing methods was also presented. The algorithm requires only the storage of the SD filter vectors in addition to the shade threshold value and can be easily implemented with a parallel algorithm. The step and ramp edge models give almost identical results, with the step model maintaining slightly higher PSNR dB values.

REFERENCES

[1] R. M. Gray, "Vector quantization," IEEE ASSP Mag., April 1984, pp. 4-29, April 1984.

[2] N. M. Nasrabadi, and R. A. King, "Image coding using vector quantization: A review," IEEE Trans. Communications, vol. COM-36, pp. 957-971, Aug. 1988.

[3] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," IEEE Trans. Communication, vol. COM-34, no. 11, pp. 1105-1115, Nov. 1986.

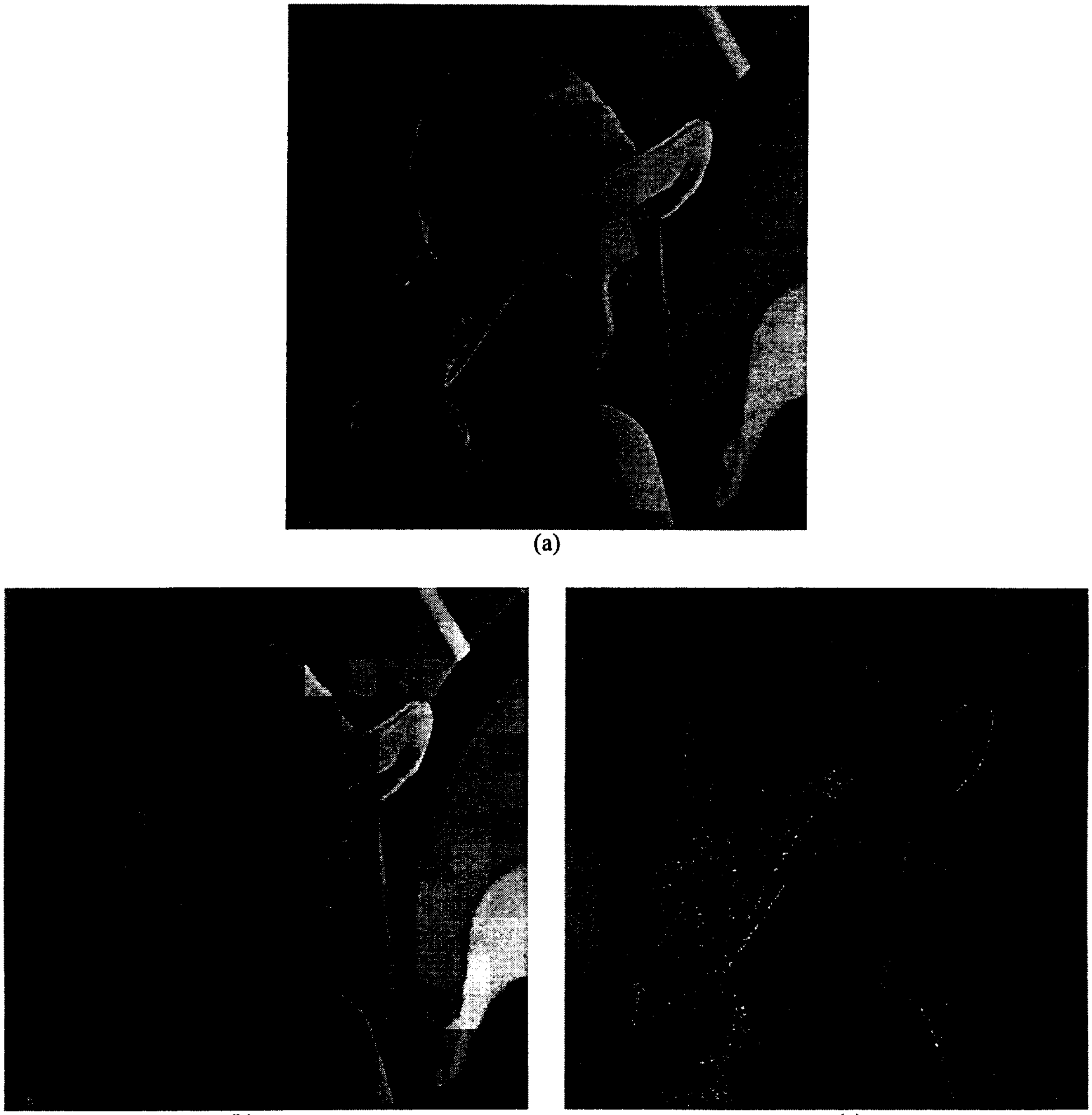
[4] J. B. Farison, Y. Shin and J. W. V. Miller, "Explicit mathematical results for the representation and filtering of spatially-invariant image sequences," Proc. of the

IEEE International Conference on Sys. Eng., Dayton, Ohio, pp. 147-153, Aug. 1991.

[5] J. W. V. Miller, J. B. Farison and Y. Shin, "Spatially-invariant image sequences," IEEE Trans. Image Processing, vol. 1, no. 2, pp. 148-161, Apr. 1992.

[6] M. K. Quweider and E. Salari, "Classification and codebook design for a vector quantizer," Proc. of SPIE Visual Communications and Image Processing Symp., vol. 2308, pp. 366-376, Sept. 1994.

[7] D. S. Kim and S. U. Lee, "Image vector quantizer based on a classification in the DCT domain," IEEE Trans. Communication, vol. COM-39, no. 4, pp. 549-556, Apr. 1991.



(a)

(b)

(c)

Figure 2. (a) Original Lena Image. (b) Reconstructed image at 0.813 bpp. (c) Difference image.