14037

p. 4

A PROGRESSIVE DATA COMPRESSION SCHEME BASED UPON ADAPTIVE TRANSFORM CODING[1]

Mixture Block Coding of Natural Images

Martin C. Rost and Khalid Sayood
Department of Electrical Engineering
University of Nebraska, Lincoln, NE 68588-0511

N92-23423

Abstract

A method for efficiently coding natural images using a vector-quantized variable-blocksized transform source coder is presented. The method, Mixture Block Coding (MBC), incorporates variable-rate coding by using a mixture of discrete cosine transform (DCT) source coders. Which coders are selected to code any given image region is made through a threshold driven distortion criterion. In this paper, MBC is used in two different applications. The base method is concerned with single-pass low-rate image data compression. The second is a natural extension of the base method which allows for low-rate progressive transmission (PT). Since the base method adapts easily to progressive coding, it offers the aesthetic advantages of progressive coding without incorporating excessive channel overhead. Image compression rates of approximately 0.5 bit/pel are demonstrated for both monochrome and color images.

1 Introduction

Natural images contain regions of high and low detail. The regions of high detail are more difficult to code than those of low detail. Since the difference between number of bits required to code these two types of regions with acceptable distortion levels can be quite large, it is desirable to use a variable-rate source coding method. One way to do this is by using a transform coder which has more than one blocksize. Regions of high detail are coded with smaller blocks, while regions of low detail are coded with larger blocks. If a similar number of bits are used to code each of the different blocksizes, variable-rate coding is achieved. When trying to maximize the quality of the reconstructed image, better usage of coding bits can be attained with the use of vector quantization. The goal is to describe a method for coding images using both vector quantization and multiple-blocksize transform coding. A block-threshold technique is used to select the blocksize used to code any particular region of the image.

Both vector quantization and transform coding are block coding methods. Block coding methods are very useful when designing low-rate image compression systems, and are used almost exclusively when coding at low rates ($<$1bit/pel). Traditional methods presented in the literature use either of these techniques, but have rarely use both together until very recently. One of the earlier publication to do this appeared in 1984 [1].

When using traditional vector quantizers for coding images, small blocksizes are used to limit the size of the required quantizer codebook. But, when coding natural images with a very small number of bits it is desirable to use as large a blocksize as possible to take maximum advantage of the high inter-pixel correlations. In general, this blocksize is usually larger than vector quantization techniques can comfortably handle. One method to overcome this problem incorporates subsampled vector quantization [2], but little has been done to use traditional vector quantization with variable-rate coding.

Small codebooks are needed because the best performing vector quantization techniques (i.e., the LBG method [3]) are clustering techniques whose codebooks are very unstructured. As a result, the codebooks are difficult to construct and use. If a codebook is designed to be less computationally intensive, such as with lattice quantizers [4], or pyramid vector quantizers [5], the attainable distortion per codeword increases for a given coding rate.

Transform coding techniques easily allow for the use of large blocksizes so high data compression ratios can be attained. But, it is difficult to keep good high-detail resolution when using large transform blocksizes [6]. This is true since most methods code only the low-frequency high-energy transform coefficients [7]. As a result, the high-frequency coefficients are often ignored. Since these coefficients carrying most of the information about the image's finer detail image quality suffers. Even when the image coefficients are vector quantized so more coefficients can be coded for a given average rate, it can still be difficult to get good high-detail resolution.

By using more than one blocksize, some of the inherent problems associated with low-rate transform coding can be overcome. Especially when vector quantizers are used to code the transform coefficients. The vector quantizer codebooks used here are of low dimensionality to keep their implementation simple. This is done by limiting the number of coefficients coded within any given block. For the examples given below, the vector quantizers code at most three coefficients as a vector for monochrome images, and nine coefficients (three transform pels) for color images. Never more than four transform pels are coded within any block, no matter is size.

After a block is coded using these coefficient-limited transform coders the distortion is measured, to see if it meets a predetermined coding threshold. If a block codes poorly, it is divided into four smaller blocks and recoded until a distortion threshold is meet or the minimum blocksize is attained. Thus, keeping the overall image integrity high by using the smaller blocks to more intensely code the high-detail regions.

In section 2 the threshold driven MBC coding algorithm is discussed. Also, the required overhead sent to the receiver to describe the final block structure of the coder is presented. Section 3 is a presentation of the MBC progressive transmission (MBC/PT) modification. In sections 4 and 5 the transform coder and the vector quantizers used in the example are shown. And finally, several examples are presented.

2 The Threshold Driven Structure of MBC

As mentioned above, each block of the image is coded using only a small number of transform coefficients. The difference between the original image and the coded image block is measured, and if the difference does not fall below a predetermined threshold, the block is divided into four smaller blocks and recoded. A new threshold is then applied to see if any of these blocks need to be divided further. This divide-and-test algorithm is continued until the entire image is coded with distortion that is less than the block threshold levels or the smallest blocksize is reached.

The monochrome images are coded using the maximum absolute difference distortion measure.

$$d = \max_i |x_i - y_i|$$

where the range of i is taken over the image block being coded, and y_i is the coded value of pixel x_i. For color images the maximum mean square difference is used,

$$d = \max_i \sqrt{(x_i - y_i)^T (x_i - y_i)/3}$$

where y_i is the coded value of color pel x_i.

---

[1] This work was supported by the NASA Goddard Space Flight Center under grant NAG 5-916.

To demonstrate the MBC method consider the coding of an example image segment. In the following it is assumed the image is coded with a starting blocksize of 16×16 and a smallest blocksize of 2×2.

First, a 16×16 block is coded, using the DCT method described below, and the distortion level for the block is measured. If this distortion is greater than the predetermined maximum level for 16×16 blocks, dmin(16×16), the block is divided into four 8×8 blocks for additional coding. After each of the 8×8 blocks is coded their resulting distortion levels are compared with the 8×8 distortion level, dmin(8×8). This process is continued until the only image blocks not meeting their given distortion threshold are those of size 2×2. Since 2×2 is the smallest allowed blocksize, these blocks are transmitted de facto, making no further attempt to improve their distortion level.

Each 16×16 block can be completely coded, using all blocksizes, before moving onto the next 16×16 block or all the 16×16 blocks of the entire image can be coded before moving to the 8×8 blocks. For MBC, this sequence is immaterial. The later method allows one to develop the progressive technique introduced in the next section.

Consider the example coding of a 16×16 image block shown in Figure 1. For clarity in the following material, let the four sub-blocks of an arbitrary block be numbered as shown in Figure 2, and let the block distortion thresholds be:

dmin(16×16)=12,
dmin(8×8)=12 and
dmin(4×4)=10.

After coding this example d(16×16) is, say, 30. So this block must be divided and recoded as four 8×8 blocks. Let the four coded 8×8 blocks have distortion levels, as shown in Figure 2, of 10, 15, 4, and 6. Since one of these 8×8 blocks fail to met dmin(8×8), it must be divided and recoded as four 4×4 blocks. Letting the 4×4 distortions levels of this 8×8 blocks be 10, 8, 15, and 6 it can be seen that only one of the 4×4 blocks fails to met dmin(4×4). This block is recoded as four 2×2 blocks.

The above coding process is depicted as a quad tree structure in Figure 3. Here there is

one 16×16 block,
four 8×8 blocks,
four 4×4 blocks and
four 2×2 blocks,

for a total of 13 blocks of coded information which are connected together through the use of a relative pointer scheme. To guarantee the receiver reconstructs the image correctly, a bit map of side information is sent along with the block coefficient information. One bit of side information is needed for each block of size greater than 2×2. If a block is to be divided, its bit value is set to 1; if not, its bit value is set to 0. To tell the receiver the ordering of these 13 coded blocks, 9 bits of side information are needed:

1,0100,0010.

The first bit shows the 16×16 block is divided into 8×8 blocks. The next four bits indicate only the second 8×8 block is divided. The last four bits indicate that only the third 4×4 block is divided into 2×2 blocks. Notice the 2×2 blocks of this example are placed with the bit maps generated at the 4×4 level, so no side information is needed to code them.

If b_i is the number of pels and d_i is the number of bits used to code a block in the i-th pass, then the average coding rate for the blocks of this pass is $r_i=(d_i+k_i)/b_i$. In the last pass (n-th) there are no overhead bits, so $r_n=d_n/b_n$. The average coding rate for the entire MBC system is

$$R=\sum_i p_i r_i, \text{ and } \sum_i p_i=1. \tag{1a,b}$$

where p_i is image fraction coded with i-th pass blocksize.

There is no reason to force the coding method of one blocksize onto another block-size, as is the case for the examples of this paper. Sometimes this maybe detrimental or, in fact, be impossible. Consider the case where blocks of size 16×16 are code with, say, six DCT coefficients. The 8×8 and 4×4 blocks could be coded in a similar fashion but, of course, it is impossible to DCT code the a 2×2 blocks with six transform coefficients. This indicates the 2×2 blocks must be coded with a different method. Likewise, as indicated above, there is no reason to keep the distortion threshold levels constant. It is even possible to change the distortion measure for each blocksize. And, since the different blocksizes encompass different spatial frequency ranges it may be desirable to do this.

3 Progressive Transmission MBC

Progressive transmission has grown out of need to transmit images over channels whose bandwidth is dramatically smaller than what is available for the timely reconstruction of full-field imagery. Since slow-scan reception of images is nonaesthetic, the need to update the image on a frame-by-frame basis (or progressive basis) has arisen. In general, most methods found in the literature are concerned with perfect reconstruction of the image. The image is transmitted on a continually improving basis until it is completely transmitted so that it can be reconstructed without error. This requires the transmission of far more data than is needed to attain a visually pleasing reconstruction of the image (as is the case considered here). But, much of this literature is directly applicable to the low-rate transmission of images since almost every PT method reconstructs a visually acceptable field within a limited number of passes [e.g., 12].

In the previous section a single an example 16×16 block was coded by passing through all of the necessary blocksizes before moving on to the next 16×16 block. But, if the entire image is passed through for each blocksize and the difference image is save for additional coding as is necessary in the next pass, the MBC method can be used as a PT coder. If each pass is immediately transmitted, the receiver can be reconstructing a crude representation of the image using these larger blocksize coefficients while the coder is processing the next pass. All passes after the first need only code the image residuals. The residuals coding information received in subsequent passes is added to the "already waiting" image of the receiver. The image is updated using smaller blocks so it acquires more clarity with each pass. Since these blocks are of smaller size, each pass updates higher-frequency image components than were coded in the previous passes.

Since the first pass is coded with very few bits the receiver has an image, although a crude image, almost immediately. Since the successive passes are coding the difference image instead of the original there is no serious problem in updating the base MBC method for PT coding.

Since only image regions that code poorly are updated in successive passes, only those regions of the image which need additional coding will continue to use coder resources. This means the regions of the image with low detail are coded quickly, and remain fixed, as the rest of the image continues to change as the information for each new pass is received.

The high detail regions of an image are coded more than once when using the MBC/PT method. Not only do the high detail regions require a greater channel capacity to transmit their coefficients because smaller blocksizes are being used but, it they also require channel resources in each of the previous passes. The rate for a MBC/PT coder is calculated using (1a), but the pass fractions are no larger constrained to add to one, in fact

$$\sum_i p_i \geq 1, \text{ and } p_i \geq p_j \text{ for } i \leq j. \tag{2}$$

Thus, applying this to (1a) shows that is it possible to have $R_{MBC/PT} \geq R_{MBC}$. But this can be offset by the fact that MBC/PT may converge to the original image more quickly and require fewer blocks since the busy sections of the image are coded with information that is taken from one than a single pass. As is shown in section 5, an MBC/PT image can require fewer coding bits to transmit than image of similar quality using the MBC method.

4 The Transform Coder

If a large block does not adequately code a given image region, it is divided into smaller blocks and recoded. So there is no strict advantage in using a large number of coefficients to code any particular blocksize. In fact, there is a tradeoff between expending more effort coding the larger blocks so fewer smaller blocks are used, and coding the larger blocks minimally so to let the threshold algorithm assign more smaller blocks for coding.

For the examples of this paper it was chosen to code each block with only four DCT transform coefficients, including the dc and three lowest order frequency coefficients (Figure 4). This was done, not so much to attain the best overall coding rate, but, to strike a median between PT coders which code a minimum of information about a given block [8] and those which code a large amount of information per block [9]. This accomplishes two things. Firstly, it shows an image can be adequately coded in a relatively small number of passes (four for the examples here) using a small number of transform coefficients at each pass, and secondly, it shows that this can be done using a simple coding algorithm for each pass. In addition, when using the same transform coder for each pass it is also possible to use share the same vector quantizer between all of the passes. This saves quantizer design effort.

5 The Quantizers

The following four paragraphs describe the quantizers used to code the examples discussed in section 5 of this paper. The remaining paragraphs discuss the details of LBG quantizers used.

When the MBC method was used to code monochrome images, the dc transform coefficients were coded with an 8-bit linear scalar quantizer (LSQ), and the ac transform coefficients were coded as a single 3-dimensional vector using an LBG vector quantizer whose codebook was of size 256.

When the MBC/PT method was used to code monochrome images, the quantizers for the first pass were different than those of subsequent passes. The these later passes code difference images that have nearly zero means blocks, while the first pass deals with the original image which is not zero mean. The dc coefficients for the first pass were coded with an 8-bit LSQ. In subsequent passes, the dc coefficients were coded with a 5-bit optimal laplacian scalar quantizer (OLSQ) [10]. The non-dc coefficients were coded the same for all passes using an LBG vector quantizer whose codebook contained 256 vectors.

The three non-dc coefficients of the YIQ images, when using the method MBC, were quantized with a vector quantizer whose codebook contained 1024 vectors. The dc Y-components were quantized with an 8-bit LSQ and the dc I- and Q- components were coded with a 5-bit OLSQ.

As with the monochrome MBC/PT method, the YIQ MBC/PT images where coded differently in the first pass than in the subsequent passes. As was the case for MBC, the dc Y-component was quantized with an 8-bit LSQ, while the I- and Q-components were quantized with a 5-bit OLSQ. In subsequent passes, the dc coefficients were quantized with an LBG vector quantizer whose codebook contained 64 vectors. The non-dc coefficients were quantized with the same LBG vector quantizer, whose codebook contained 1024 vectors, for all passes.

It was decided to use the same vector quantizer for the ac transform coefficients of each pass, no matter the blocksize. This rule was chosen because the ac coefficients for all passes were found to be distributed in a similar fashion. The only adjustment to be made was to allow for the differences in the coefficient variances, which had to be adjusted for each pass to guarantee a universal match the codebook variance.

By using this variance matching technique, more training vectors could be taken from a single image, so fewer images were needed to develop an adequate training set. The training set was built using scaling factors that mapped the unquantized ac coefficients of each pass into unit pdf. The scaling factors used are shown in Tables 1 and 2. The final vector quantizer codebooks, as was indicated above, held 1024 9-dimensional vectors for the YIQ coders and 256 3-dimensional vectors for the monochrome coders. They were built using a training set taken from three images different from the one coded for this paper.

The final vector quantizers were chosen to be nonadaptive. This is, the codebook was not modified to more effectively quantize out-of-bound source vectors as the coder moves from image to image [e.g., 11]. This was done for two reasons. Sometimes the overhead required to implement such a technique can be overly expensive and the return acquired from it minimal. The extra effort needed stood against the design goal to construct a "simple to implement" vector-quantized adaptive transform source coder.

6 Results

All of the examples, as listed in Tables 3-6, use the 512×512 RGB woman/hat picture of the UCLA database. The monochrome examples use the green (G) color field, while the YIQ images are made using the RGB to YIQ transformation matrix of [12]. All the examples use a starting blocksize of 16×16 and a final blocksize of 2×2. These tables list the number of blocks coded for each blocksize, and the thresholds used to test the quality of the coding passes. Also, the MBC/PT tables list the average coding rate that has accumulated after each pass.

This rate is based upon the average of the coding bits as spread across the entire image, without concern for what fraction of the image is coded within any particular pass. These rates represent the coding rate that is required to code the image if the coder where to stop with that particular pass. Since the remaining passes are yet to be coded, the image percentage coded within the indicated pass must be updated to include the image percentages coded in all of subsequent passes. For example, consider the MBC/PT rate of Table 4 when stopping at 4×4 blocks. In this case, the rate is .358 bits/pel, and the percentage of blocks coded with 4×4 blocks is 25.25 percent.

To code the monochrome image with MBC/PT only requires an extra .011 bits/pel over MBC. The overhead needed to code any given image is a function of the LBG codebook, and the codebook is a function of the training set used. A differently constructed codebook could offer different results. It is interesting to note that the YIQ MBC/PT method requires less coding rate to obtain the same image quality (PSNR) as is obtained when using MBC alone. It is clear that MBC/PT does not require excessive overhead to add the desirable PT feature.

References

[1] K. Sayood, J. Gibson and M. Rost, "An algorithm for uniform vector quantization design," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 6, pp. 805-814, Nov. 1984.

[2] H.-M. Hang and B. G. Haskell, "Interpolative vector quantization of color images," *IEEE Trans. Commun.*, vol. COM-36, pp. 465-470, Apr. 1988.

[3] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[4] J. D. Gibson and K. Sayood, "Lattice quantization," to appear in *Advances in Electronics and Electron Physics*.

[5] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 568-583, July 1986.

[6] D. J. Vaisey and A. Gersho, "Variable block-size image coding," *Proc. ICASSP* 87, vol. 2, pp. 1051-1054, Apr. 1987.

[7] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[8] K. Knowlton, "Progressive transmission of grey-scale and binary pictures by simple, efficient, and lossless encoding schemes," *Proc. IEEE*, vol. 68, pp. 885-896, July 1980.

[9] E. DuBois and J. L. Moncet, "Encoding and progressive transmission of still pictures in NTSC composite format using transform domain methods," *IEEE Trans. Commun.*, vol. COM-34, pp. 310-319, Mar. 1986.

[10] W. C. Adams and C. E. Gissler, "Quantizing characteristics for signals having laplacian amplitude probability density functions," *IEEE Trans. Commun.*, vol. COM-26, pp. 1295-1297, Aug. 1978.

[11] D. J. Vaisey and A. Gersho, "Variable blocksize image coding," *Proc. ICASSP*, Apr. 1987, pp. 1051-1054.

[12] L. Wang and M. Goldberg, "Progressive image transmission by transform coefficient residual error quantization," *IEEE Trans. Commun.*, vol. 36, pp. 75-87, Jan. 1988.

[13] W. K. Pratt, *Digital Image Processing*, New York, NY: Wiley, 1978.

Table 1.
Vector quantizer scale factors for monochrome images.

Blocksize	scale factor
16×16	60
8×8	35
4×4	20
2×2	10

Table 2.
Vector quantizer scale factors for YIQ images.

Blocksize	scale factor
16×16	60
8×8	35
4×4	20
2×2	10

Table 3.
Monchrome MBC rate (bits/pel)

min blocksize	#blocks	%image	PSNR	rate	threshold
16×16	533	52.05	23.11 dB	-	33
8×8	1048	25.59	26.22	-	33
4×4	2950	18.01	29.61	-	33
2×2	2856	4.36	31.01	.479	-

Table 4.
Monchrome MBC/PT rate (bits/pel)

min blocksize	#blocks	%image	PSNR	rate	threshold
16×16	1024	100.00	23.13 dB	.063	33
8×8	1940	47.48	26.42	.170	33
4×4	3520	21.48	29.76	.358	33
2×2	2472	3.77	30.94	.490	-

Table 5.
YIQ MBC rate (bits/pel)

min blocksize	#blocks	%image	PSNR	rate	threshold
16×16	590	57.62	24.09 dB	-	12
8×8	976	23.83	26.77	-	12
4×4	2444	14.92	29.06	-	11
2×2	2384	3.64	29.69	.691	-

Table 6.
YIQ MBC/PT rate (bits/pel)

min blocksize	#blocks	%image	PSNR	rate	threshold
16×16	1024	100.00	24.12 dB	.110	12
8×8	1760	42.97	26.89	.220	12
4×4	2870	17.52	29.69	.402	11
2×2	1820	2.78	29.63	.516	-

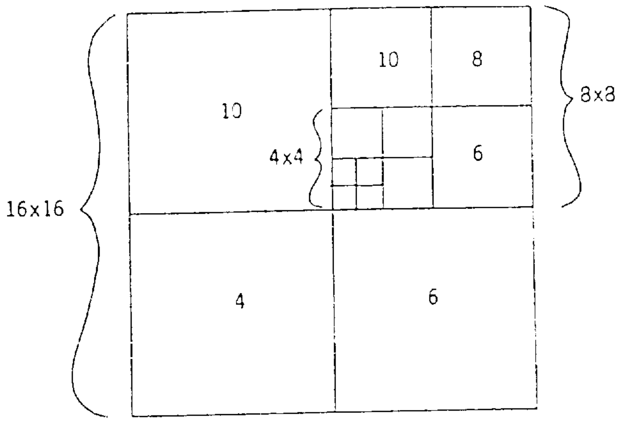

Figure 1. Example coded 16x16 block

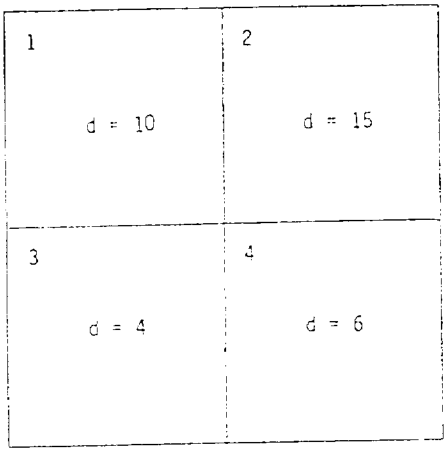

Figure 2

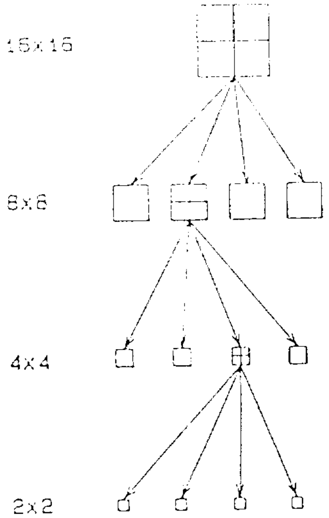

Figure 3. Example Tree Structure

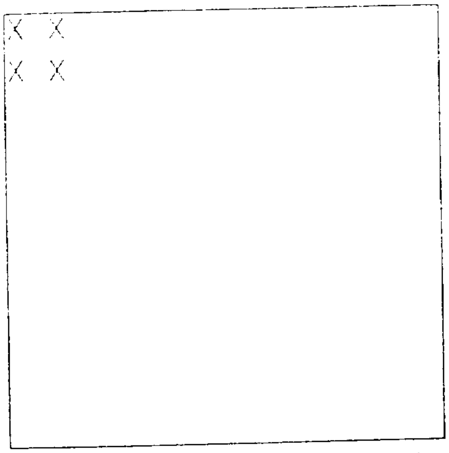

Figure 4. Lowest order DCT coefficients