Efficient Variable Rate Vector Quantization Using Quadtree Segmentation

Hung Yuen and Kam-Chi Li

Department of Electronic Engineering
Hong Kong Polytechnic
Hung Hom, Hong Kong

Lajos Hanzo

Dept. of Electronics and Computer Science
University of Southampton
SO9 5NH, UK

Abstract—**Variable rate vector quantization (VQ) using greedy tree growing algorithm has been used in image compression. In previous work, images have been partitioned into fixed size blocks and therefore correlation existing in large variable size blocks cannot be exploited. In this paper, a new and efficient variable rate vector quantization scheme using variable block sizes is introduced. Simulation results show that the proposed scheme which is based on variable block size quadtree segmentation can achieve better performance than the fixed block size variable rate VQ.**

I. INTRODUCTION

Vector quantization is a data compression technique that has been widely used for image compression. The performance of a vector quantizer can be improved by using a variable rate code [1] which is able to designate more bits to regions of an image that are more active or difficult to code and fewer bits to less active regions.

Greedy tree growing algorithm [2] is a technique for directly designing a variable rate tree-structured vector quantizer (TSVQ) by growing the tree one node at a time, rather than one layer at a time as in the case of a fixed rate TSVQ. It results in an unbalanced tree since the node that is being split can be at any depth. The tree is grown until the average rate of the unbalanced tree equals some target rate. The variable rate code for a particular codeword is just its path map on the unbalanced tree.

The greedy tree growing algorithm finds the node that when split, it yields the largest decrease in distortion per increase in average length, i.e., best trade-off rate and distortion in a greedy sense. Therefore, the greedy unbalanced tree can code high distortion regions at a higher resolution since it is not limited to a maximum depth. However, such desirable property occurs only in the codebook design, but not in the encoding process.

Fixed block size variable rate VQ has limitation that it does not take advantage of correlation in large variable size regions which have low activity. For images with large homogeneous background regions, further compression is possible by using variable size blocks.

Quadtree segmentation [3] becomes a new direction in image coding, which partitions an image into variable size regions based on a quadtree structure. Studies have demonstrated that quadtree-based image segmentation can be an effective and efficient mechanism for isolating blocks of distinct perceptual significance and thereby allowing different coding strategies that are perceptually suited to the individual segment categories. It provides an effective compromise between the accuracy with which the region boundaries are determined and the number of bits required to specify the segmentation information. Small overhead is required due to its restriction on the shape and the number of possible sizes of the final regions. In performing such a segmentation, the coding algorithm must take a series of binary decisions that are controlled by a number of different threshold values.

However, Vaisey and Gersho [3] used quadtree segmentation only in the encoding process, but not in the quantizer design, where they employed classified VQ to encode the small high-detail regions. To exploit correlation between large variable size blocks existing in training set and testing images, we propose to apply quadtree

0-7803-2570-2/95 $4.00

segmentation both in the quantizer design and the encoding process. The segmentation is performed with a quadtree structure by isolating the more important regions into small regions and separately identifying large low detail regions. Since the important regions have been isolated, the remaining parts of the image can be coded at a much lower rate.

II. QUADTREE SEGMENTED VARIABLE RATE VQ

To obtain the flexibility of a variable block size image segmentation while avoiding the excessive overhead needed to specify its partition, we use the top-down quadtree approach.

A quadtree is a tree structure in which each nonterminal node has four branches emanating from it. These branches point to nodes that are the children of the parent node. In our implementation, every node corresponds to a subblock of the image that is determined, both in size and location, by its position in the tree. The four children of a particular parent node represent the four subblocks obtained by splitting the parent block into four equal-sized squares. The tree consists of several levels of nodes, where a node at the nth level represents a one-quarter subblock of the parent image block from the previous level, as shown in Fig. 1.

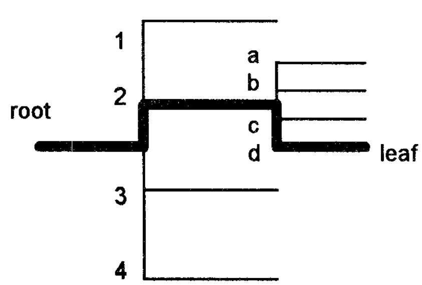

Fig. 1 Quadtree structure

Since analysis of images showed that the means of blocks larger than 32×32 are almost always inhomogeneous, we start with a segmentation of images into blocks of this size. The variance of each block is used as the threshold function. For each block, its block mean and variance are calculated. If the variance is below some predetermined value, this block is declared as a leaf block. Otherwise, the block is further divided into four smaller blocks. This process repeats recursively until it reaches block size of 4×4. For images in our implementation, block sizes that are used are 32×32, 16×16, 8×8 and 4×4.

In the vector quantizer design, we use quadtree segmentation to partition the training set so that all the high detail regions are isolated into the 4×4 blocks, which is used to design the codebook using the greedy tree growing algorithm. For the encoding process, quadtree segmentation is used to partition the test image into high detail 4×4 blocks and low detail large blocks. The high detail 4×4 blocks are encoded by the VQ codebook generated by greedy tree growing algorithm. The low detail large blocks are encoded by their block means to achieve further compression.

Since variable block sizes are used in quadtree segmentation, decoding of transmitted images requires the information about the size and location of each block. If scalar quantization is used to quantize such information, two bits are required to represent the block size and twelve bits are required to indicate the position of the block when the image to be coded is 256×256. This amounts to too much overhead information needed for transmission.

In our implementation, a novel and efficient scheme based on the quadtree structure is used to indicate each block size and its location. Sequentially starting from 32×32 blocks, for each block, the first bit is used to indicate whether the block is a leaf block or a node block. If it is a leaf block, this bit is followed by its quantized block mean. If it is a node block, that block must have been partitioned into four smaller blocks and so we repeat the previous step until we reach all the 4×4 blocks, which are encoded by the vector quantizer using greedy tree growing algorithm. Therefore, our scheme uses less than three bits on average to indicate each block size and its location.

In designing the tree-structured codebook, we use an efficient and faster splitting algorithm [4], other than the Riskin and Gray's approach [2, 5] which requires a much more complex one step look-ahead splitting. Due to quadtree segmentation, training vectors for designing codebook are more compactly located and therefore the computationally intensive look-ahead splitting is not needed.

Let D be the total distortion contributed by a particular node and N be the number of vectors within that node. The goodness of a split among all the terminal nodes is determined by $\lambda = D/N$. That is, we split the node that contributes the largest average distortion. The unbalanced tree is grown until its average rate equals the target rate. The transmitted index of a particular vector is the binary sequence indicating the path from the root of the tree to that leaf.

III. SIMULATION RESULTS

To evaluate the performance of the proposed quadtree segmented variable rate VQ using greedy tree growing algorithm, several test images of size 256×256 were coded and compared to the conventional greedy tree growing VQ [2, 5]. The initial block size for quadtree segmentation was set to 32×32 and the smallest block size was 4×4. The high detail 4×4 blocks were encoded by variable rate VQ using the efficient greedy tree growing algorithm [4] and the larger blocks were encoded by their mean values.

Fig. 2 shows the original image "Lenna" and Fig. 3 shows the reconstructed image using the proposed quadtree segmented greedy tree growing VQ. The PSNR of the proposed scheme at an average rate of 0.3633 bpp is 28.08 dB whereas in Fig. 4 it is 26.23 dB for the conventional approach using fixed size blocks at 0.375 bpp. This represents a gain of 1.85 dB. Clearly application of quadtree segmentation in the training set and the encoding process improves the PSNR performance.

For images with large low detail regions, gain in PSNR is more remarkable. Fig. 5 shows the original image "House". Fig. 6 and Fig. 7 show the reconstructed images using proposed scheme and conventional method, respectively. A gain of 2.19 dB is achieved. It shows that additional compression can be achieved by using variable block sizes to take advantage of large variable size image contents. It can be observed clearly that image quality of high detail regions and particularly, edges has been improved using the proposed scheme, although some low detail regions seem to be slightly more blocky.

IV. CONCLUSIONS

Since our approach uses quadtree segmentation to extract the high detail regions from the training set, it reduces the codebook design time as compared to Riskin and Gray's approach by avoiding splitting nodes in those low detail regions. It achieves better performance because of the segmentation in the training set which tailors the codebook to the high detail regions, and the segmentation in the input images which concentrates the coding power in the regions where it is most needed. The codebook become more specialized in its task due to the fact that the training set is no longer corrupted by many low detail vectors.

Since the greedy tree growing algorithm has the ability of coding high distortion regions at a higher resolution, our approach has less complexity as compared to the Vaisey and Gersho's one [3] which uses classified VQ. It is able to exploit correlation between large variable size blocks existing in training set and in testing images as well.

Another advantage of this scheme is that it is flexible in the sense that image quality can be gracefully degraded for lower bit rates with no further increase in complexity. If low bit rate is required, a large threshold can be used during the input image segmentation. It results in more coarsely encoded large blocks and degraded image quality, without modifying the vector quantizer.

REFERENCES

[1] T. Lookabaugh, E. A. Riskin, P. A. Chow, and R. M. Gray, "Variable rate vector quantization for speech, image, and video compression," *IEEE Trans. Communication*, vol. COM-41, pp. 186-199, Jan. 1993.

[2] E. A. Riskin and R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizer," *IEEE Trans.Signal Processing*, vol. 39, pp. 2500-2507, Nov.1991.

[3] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 2040-2060, Aug. 1992.

[4] W. J. Zeng, Y. F. Huang, and S. C. Huang, "Greedy tree growing algorithms for designing variable rate vector quantizers," *Proc. ISCAS*, vol. V, pp. 594-597, 1993.

[5] E. A. Riskin and R. M. Gray, "Lookahead in growing tree-structured vector quantizers," *Proc. ICASSP*, vol. 4, pp. 2289-2292, 1991.

Fig. 2 Original image "Lenna" at 8 bpp.

Fig. 3 "Lenna" coded with proposed scheme at 0.3633 bpp, PSNR is 28.08 dB.

Fig. 4 "Lenna" coded with greedy tree growing VQ at 0.375 bpp, PSNR is 26.23 dB.

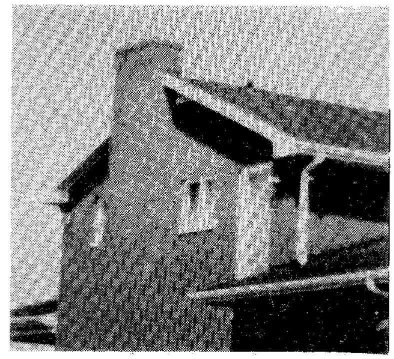

Fig. 5 Original image "House" at 8 bpp.

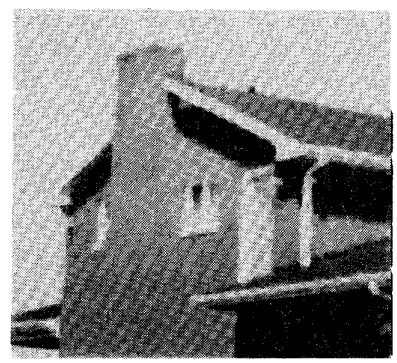

Fig. 6 "House" coded with proposed scheme at 0.2312 bpp, PSNR is 27.73 dB.

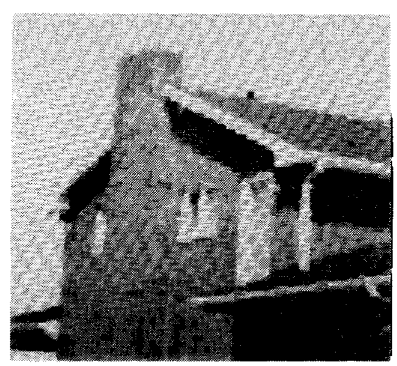

Fig. 7 "House" coded with greedy tree growing VQ at 0.25 bpp, PSNR is 25.54 dB.