LOCALLY OPTIMAL CODEBOOK DESIGN FOR QUADTREE-BASED VECTOR QUANTIZATION

Michael Lightstone, Kenneth Rose, and Sanjit K. Mitra

Center for Information Processing Research
Department of Electrical and Computer Engineering
University of California, Santa Barbara 93106
(mlight@ece.ucsb.edu)

ABSTRACT

The optimal design of quadtree-based vector quantizers is addressed. Until now, work in this area has focused on optimizing the quadtree structure for a given set of leaf quantizers with little attention spent on the design of the quantizers themselves. In cases where the leaf quantizers were considered, codebooks were optimized without regard to the ultimate quadtree segmentation. However, it is not sufficient to consider each problem independently, as separate optimization leads to an overall suboptimal solution. Rather, joint design of the quadtree structure and the leaf codebooks must be considered for overall optimality. The method we suggest is a "quadtree" constrained version of the entropy-constrained vector quantization design method. To this end, a centroid condition for the leaf codebooks is derived that represents a necessary optimality condition for variable-rate quadtree coding. This condition, when iterated with the optimal quadtree segmentation strategy of Sullivan and Baker results in a monotonically descending rate-distortion cost function, and consequently, an (at least locally) optimal quadtree solution.

1. INTRODUCTION

Natural images typically consist of regions with widely varying content and activity that often frustrate coding efforts. Because of these nonstationarities, it is desirable to find a segmentation that allocates less bits to homogeneous neighborhoods and more bits to areas containing edges and texture. When this strategy is adopted, the segmentation structure along with the quantization information must be specified to the decoder in order to reconstruct the coded image. One widely used and successful segmentation scheme that can be easily transmitted due to its compact nature is the quadtree data structure [1].

The quadtree data structure is a method for hierarchically decomposing an image into distinct, nonoverlapping regions of varying dimension. For example, an image block of size $M \cdot 2^L \times M \cdot 2^L$ can be decomposed into an L level hierarchy ranging from 4^L leaf nodes of size $M \times M$ to a single leaf of dimension $M \cdot 2^L \times M \cdot 2^L$ (see Figure 2). When used for coding, an image is quantized using a set of codebooks matched to the dimensions of the leaf nodes. Until recently, the determination of the quadtree decomposition has been limited to methods that are heuristic in nature [2, 3, 4, 5]. For example, in the ***top-down*** (splitting) approach [3], starting from the largest possible block, a predefined rule is employed to determine whether a given block should be quantized using a single quantizer, or whether it should be decomposed into four smaller subblocks to be quantized independently. In contrast, the ***bottom-up*** (merging) construction [2, 4, 5] begins with the smallest possible blocks. Using a predefined criteria, subsequent judgments are made whether any adjacent four subblocks should be merged into a single block to be quantized instead—given that all four subblocks have been previously merged.

While both of these techniques are intuitively appealing, they do *not* determine the optimal quadtree structure in the sense that overall distortion is minimized subject to a constraint on the overall rate. This difficulty was recently surmounted by Sullivan and Baker [6] in which a Lagrangian formulation similar to that of the generalized BFOS algorithm [7] was employed to determine the optimal structure for a given set of quantizers—including the overhead information to specify the tree. Using the nested nature of the quadtree segmentation, this technique eliminates the necessity of exhaustively searching over all possible quadtree structures.

However, the optimality of this approach is limited by the quality of the leaf codebooks. In [6], as with most previous methods, codebooks for each of the varying dimensions are designed without regard for their role as part of the overall quadtree structure. For cases in which an attempt is made to reflect the role of the codebooks, they are heuristic and consequently suboptimal [3]. For the best results, leaf codebooks should reflect the class of the image to which they are called on to represent. Intuitively (and as evidenced by experimental results), we expect that larger block-size codebooks should represent smooth sections, while codebooks for smaller blocks should be targeted for high activity regions involving edges and texture. In any event, for overall optimality, joint design of the quadtree structure and the leaf codebooks must be conducted. To this end, we propose an iterative procedure in Sections 2 and 3 for the design of variable block-size quantizers (and their respective variable length codes) that produces a lo-

This material is based on work supported under a National Science Foundation Graduate Fellowship and under grant No. NCR-9314335, and by a University of California MICRO grant with matching supports from Digital Instruments, Rockwell International, Tektronix, Echo Speech Corporation, Signal Technology, Inc., Qualcomm, Inc., and Lockheed Missiles and Space Company.

0-7803-2431-5/95 $4.00 © 1995 IEEE

cally optimal quadtree solution. This solution is achieved by jointly considering the design of both the leaf codebooks and the quadtree structure, and viewing the overall problem in an entropy-constrained product code framework [8]. Furthermore, in Section 4 we provide experimental evidence to demonstrate that the optimal quadtree structure for these codebooks can provide significant rate-distortion improvement over codebooks designed without considering the quadtree decomposition.

2. OPTIMAL QUADTREE CODING

Given a subimage of size $M \cdot 2^L \times M \cdot 2^L$, quadtree coding can be viewed as vector quantization (VQ) with a single *effective* codebook C. This effective codebook is constructed using variable size code vectors as building blocks from a set of L *leaf* codebooks $\{C_i;\ i = 1, \ldots, L\}$. Moreover, the leaf codebook C_i for $i = 1, \ldots, L$, consists of $M_i \times M_i$ dimensional code vectors, where $M_i = M \cdot 2^{L-i+1}$. As a result, each element (or code vector) in the effective codebook C corresponds to a particular quadtree tiling of the subimage using variable dimension code vectors selected from the leaf codebooks of appropriate size.

Now, consider a given set of image training data T, partitioned into K blocks of size $M \cdot 2^L \times M \cdot 2^L$. With regard to optimal quantization, our objective is to minimize the total distortion D using a structurally constrained, quadtree codebook C subject to a constraint on the overall rate R. In terms of entropy coding, we assume a one-to-one mapping from every code vector $\mathbf{y}$ in the leaf codebooks $\{C_i\}$ to a uniquely decodable variable length codeword whose length is specified by $l(\mathbf{y})$. For notational simplicity, we repartition the original training data T into smaller blocks of size $M_i \times M_i$ for $i = 1, \ldots, L$, resulting in L sets given by T_i for $i = 1, \ldots, L$, respectively. Then, using an unconstrained Lagrangian formulation [9], we can write the overall cost function $J(C, S) = D(C, S) + \lambda R(S)$ as:

$$J = \sum_{i=1}^{L} \sum_{\mathbf{y} \in C_i} \sum_{\mathbf{x} \in T_i} \left[S_{\mathbf{x},\mathbf{y}} \left[d(\mathbf{x}, \mathbf{y}) + \lambda l(\mathbf{y}) \right] \right] + \lambda R_t(S), \tag{1}$$

where $R_t(S)$ corresponds to the cost of transmitting the quadtree structure, itself, for all vectors in the training data. As such, we must perform the minimization over all possible structurally constrained quadtree codebooks, all variable length codes, and the set of valid selector functions given by $S = \{S_{\mathbf{x},\mathbf{y}}\}$. In our case, the selector functions are specified by

$$S_{\mathbf{x},\mathbf{y}} = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is coded with } \mathbf{y} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Using this paradigm, the algorithm for determining the best quadtree structure developed in [6] can be likened to the encoder optimality condition for VQ [10] in which the structurally constrained form of the quadtree codebook permits optimal encoding without an exhaustive search over all vectors in C. As part of this paper, we reformulate this encoder optimality condition in the product code framework and derive an analogous condition for the decoder that leads to optimal leaf codebooks and variable length codes for a given quadtree structure. The subsequent iteration of these two conditions is the foundation of a joint design algorithm that leads to a monotonically descending Lagrangian cost function in a manner resembling the generalized Lloyd algorithm [11].

2.1. Encoder Optimality

From the encoder optimality condition in [9, 10], it follows that for a fixed codebook C, the optimal quadtree representation is determined by the following Lagrangian-modified, nearest neighbor mapping:

$$Q(\mathbf{x}) = \operatorname*{argmin}_{y \in C} \left[d(\mathbf{x}, \mathbf{y}) + \lambda \cdot l(\mathbf{y}) \right]. \tag{3}$$

Here, the code vectors $\mathbf{y}$ are taken from the effective codebook C, and as such, the rate term $l(\mathbf{y})$ now includes the binary representation of the quadtree in addition to the specification of all appropriate leaf code vectors. Without regard to the constraints imposed by the quadtree structure, the necessary condition in Eq. (3) implies an exhaustive search over all code vectors in C. It is not difficult to see that even for moderately sized leaf codebooks and a small quadtree, the dimension of the effective codebook can grow quite large. For example, a 4 level quadtree with leaf codebooks of size $N_i = 4$ for all i corresponds to an effective codebook with more than 4×10^{38} code vectors. Fortunately, because of the nonoverlapping nature of the quadtree segmentation, an alternate strategy exists for obtaining the optimal code vector in C with very manageable complexity [6].

First, consider the following iterative construction of the effective codebook C in which we observe that $C_1^E \equiv C$ can be viewed as the union of two smaller codebooks C_1 and C_1^P. The codebook C_1, as defined earlier, consists of unstructured $M_1 \times M_1$ vectors and is used to quantize the block if it is a leaf node. If not, the block is quantized using a product codebook C_1^P which is simply the Cartesian product of four reduced dimensionality codebooks with code vectors of size $M_i/2 \times M_i/2$, or equivalently, $M_{i+1} \times M_{i+1}$. Proceeding in a recursive fashion, we can further subdivide each of the reduced dimensionality codebooks so that at level i we have an effective codebook given by $C_i^E = C_i \cup C_i^P$ and a product codebook given by $C_i^P = C_{i+1}^E \times C_{i+1}^E \times C_{i+1}^E \times C_{i+1}^E$ for $i = 1, \ldots, L-1$. At level L, the effective codebook can no longer be decomposed, and consequently, we have $C_L^E = C_L$. Our goal is to use this structure to reduce the encoding complexity.

Observe that any particular vector $\mathbf{x} \in T_i$ can be decomposed into four vectors $\mathbf{x}_j \in T_{i+1}$ for $j = 1, \ldots, 4$. In order to quantize $\mathbf{x}$ using C_i^E, we note that by construction, the optimal code vector $\mathbf{y}_E^*(\mathbf{x})$ can be determined by simply comparing the best code vector $\mathbf{y}_l^*(\mathbf{x})$ in the leaf codebook C_i with the best code vector $\mathbf{y}_P^*(\mathbf{x})$ in the product codebook C_i^P. Moreover, because of its nonoverlapping nature, the best code vector in C_i^P is just the Cartesian product of the four best code vectors in C_{i+1}^E for $\mathbf{x}_j$ ranging from $j = 1, \ldots, 4$. Hence, we choose $\mathbf{y}_l^*(\mathbf{x})$ iff

$$d(\mathbf{x}, \mathbf{y}_\ell^*(\mathbf{x})) + \lambda \cdot [l(\mathbf{y}_\ell^*(\mathbf{x})) + 1] < \sum_{j=1}^{4} \Big[d(\mathbf{x}, \mathbf{y}_E^*(\mathbf{x}_j)) + \lambda \cdot [l(\mathbf{y}_E^*(\mathbf{x}_j)) + 1] \Big], \quad (4)$$

which corresponds to updating the selector functions according to $S_{\mathbf{x},\mathbf{y}_\ell^*(\mathbf{x})} = 1$, and $S_{\mathbf{x},\mathbf{y}} = 0$ for all leaf code vectors $\mathbf{y} \in C_i^P$. Otherwise, we choose $\mathbf{y}_P^*(\mathbf{x})$ and set $S_{\mathbf{x},\mathbf{y}_\ell^*(\mathbf{x})} = 0$. Note that the additional bit in the rate term of Eq. (4) corresponds to the cost of transmitting the tree structure—assuming the quadtree is coded as in Figure 2. Using this methodology, the optimal mapping described by Eq. (3) can be achieved through a recursive bottom-up procedure in which the training data is coded using each C_i^E for $i = L, L-1, \ldots, 1$, according to Eq. (4). As a result, the total encoding complexity is reduced from a single infeasible search over the entire effective codebook, to a number of independent searches over much smaller leaf codebooks.

2.2. Decoder Optimality

We now develop a necessary optimality condition that can be used to find the best codebook C for a fixed quadtree segmentation. Consider the overall cost function given by Eq. (1). Since the codebook C is constructed from the set of leaf codebooks $\{C_i\}$, we take the partial derivative of J with respect to an arbitrary leaf code vector $\mathbf{y} \in C_i$ to arrive at

$$\frac{\partial J}{\partial \mathbf{y}} = \sum_{\mathbf{x} \in T_i} 2 \cdot (\mathbf{x} - \mathbf{y}) \cdot S_{\mathbf{x},\mathbf{y}} \quad (5)$$

when the distortion measure $d(\mathbf{x}, \mathbf{y})$ is the squared error given by $\|\mathbf{x} - \mathbf{y}\|^2$. By setting this to zero, we infer the following necessary condition for the optimal leaf code vectors:

$$\mathbf{y}^* = \frac{\sum_{\mathbf{x} \in T_i} \mathbf{x} \cdot S_{\mathbf{x},\mathbf{y}}}{\sum_{\mathbf{x} \in T_i} S_{\mathbf{x},\mathbf{y}}}, \quad \forall\, \mathbf{y} \in C_i,\ i = 1, \ldots, L. \quad (6)$$

Thus, each reproduction codeword is simply the centroid of all training vectors of appropriate dimension that are quantized by it.

Finally, for a fixed codebook and quadtree, we seek variable length codewords that minimize the total rate in Eq. (1). Under the assumption that the first-order entropy for each leaf codebook can be attained, we assign the codeword lengths according to $l^*(\mathbf{y}) = -\log_2(p_i(\mathbf{y}))$ for all $\mathbf{y} \in C_i$ from $i = 1, \ldots, L$, where $p_i(\mathbf{y})$ is the probability mass function (pmf) given by

$$p_i(\mathbf{y}) = \frac{K \cdot 4^{i-1}}{\sum_{\mathbf{x} \in T_i} S_{\mathbf{x},\mathbf{y}}}. \quad (7)$$

3. CODEBOOK DESIGN

Having established necessary requirements for a quadtree encoder and decoder in Section 2, we can update an arbitrary quadtree structure through successive iteration of the two conditions. From the necessary conditions, it follows that each step of this approach can only improve or leave unchanged the rate-distortion performance of the encoder or decoder, resulting in monotonic descent of the Lagrangian cost function, and, consequently, a locally optimal quadtree solution. For different values of λ, each solution corresponds to a distinct, but optimal, rate-distortion tradeoff. The resulting algorithm, which can be viewed as "quadtree-constrained" version of the entropy-constrained VQ design algorithm in [9], is described in Figure 1.

Step (1): Initialize:

- L leaf codebooks, C_i, $i = 1, \ldots, L$,
- L pmf's, p_i, $i = 1, \ldots, L$,
- Lagrange multiplier λ,
- Training data, $T = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$,
- A stopping threshold ϵ,
- and $t = 1$ and $J^{(0)} = \infty$.

Step (2): Determine optimal quadtree structure using Eq. (4) for $i = L, L-1, \ldots, 1$.

Step (3): Update $\{C_i\}$ according to Eq. (6).

Step (4): Update $\{p_i\}$ according to Eq. (7).

Step (5): Compute $J^{(t)}$ using Eq. (1)

Step (6): If $J^{(t-1)} - J^{(t)} < \epsilon$, then go to **Step** (8).

Step (7): Set $t = t + 1$. Go to **Step** (2).

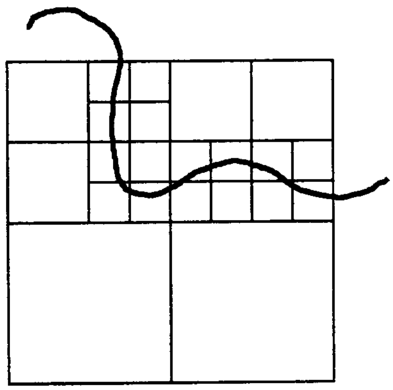
Step (8): End.

Figure 1: Quadtree design algorithm.

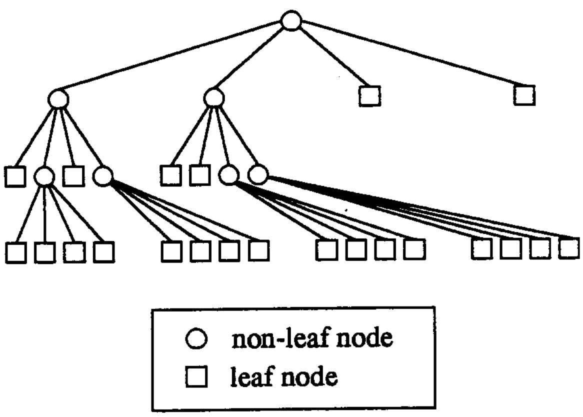
4. RESULTS

In terms of the leaf quantizers, structural constraints can be imposed to further reduce the memory requirements and encoding complexity of the overall quadtree structure. For example, we can employ variable rate product codes such as transform coding, or, as in our experiments, mean-gain-shape VQ [12], and still jointly optimize the entire structure with little modification to the proposed algorithm.

Computer simulations comparing the quadtree coding performance of three distinct codebook design approaches are now examined. All results are for the 512×512 "lena" image using a six level quadtree structure with $M = 1$. Maximum codebook sizes of the shape features for levels 1 through 6 are 1, 512, 512, 256, 128, and 32, respectively. First, we consider the approach of Sullivan and Baker in which the leaf codebooks are optimized for fixed-rate VQ independently of the quadtree segmentation. The performance using these codebooks corresponds to the bottom curve in Figure 3. Since quadtree coding is inherently variable rate, we next drop the fixed-rate constraint on the leaf quantizers, and design the leaf codebooks using entropy-constrained VQ [9]. These results correspond to the middle curve in Figure 3. We note that while the leaf codebooks and the quadtree are designed for the same value of λ, the codebook optimization is performed independently.



(a)



(b)

Code: 1 1100 0101 0011

(c)

Figure 2: Example three-level quadtree data structure: (a) quadtree-based segmentation of an image block, (b) corresponding tree structure, and (c) binary code to represent tree (0=leaf node, 1= non-leaf node).

Although this approach affords us some gain in the rate-distortion sense, better results are still possible if we implement the joint design methodology described in Section 3. The curve representing this design approach is the top one in Figure 3.

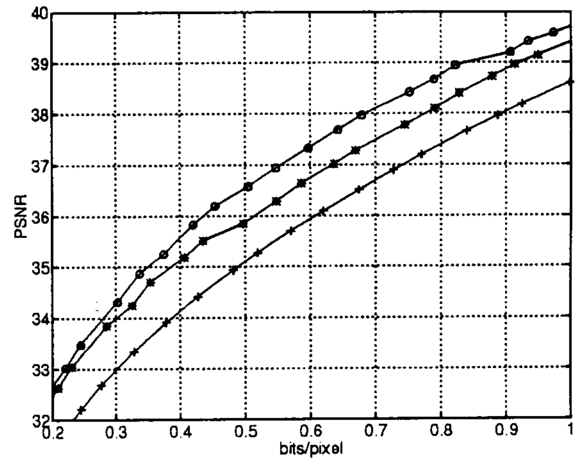


Figure 3: Quadtree coding results for leaf codebooks designed using (a) fixed rate VQ (crosses), (b) ECVQ (stars), and (c) the proposed joint design strategy (circles).

5. REFERENCES

[1] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys*, vol. 16, pp. 188–260, June 1984.

[2] P. Strobach, "Quadtree-structured recursive plane decomposition coding of images," *IEEE Trans. on Signal Processing*, vol. 39, pp. 1380–1397, June 1991.

[3] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. on Signal Processing*, vol. 40, pp. 2040–2060, Aug. 1992.

[4] L. Berger, J.P. Mariot, and C. Launay, "A new formulation for fast image coding using quadtree representation," *Pattern Recognition Letters*, vol. 13, pp. 425–432, June 1992.

[5] E. Shusterman and M. Feder, "Image compression via improved quadtree decomposition algorithms," *IEEE Trans. on Image Processing*, vol. 3, pp. 207–215, March 1994.

[6] G.J. Sullivan and R.L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. on Image Processing*, vol. 3, pp. 327–331, May 1994.

[7] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Optimal pruning with application to tree-structured source coding and modeling," *IEEE Trans. on Information Theory*, pp. 299–315, March 1989.

[8] M. Lightstone, D. Miller, and S.K. Mitra, "Entropy-constrained product code vector quantization with application to image coding," in *Proceedings of the First IEEE International Conference on Image Processing*, vol. I, (Austin, Texas), pp. 623–627, Nov. 1994.

[9] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 37, pp. 31–42, Jan. 1989.

[10] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer Academic Publishers, 1992.

[11] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. 28, pp. 84–95, Jan. 1980.

[12] M. Lightstone and S.K. Mitra, "Entropy-constrained, mean-gain-shape vector quantization for image compression," in *Proc. of the SPIE Symposium on Visual Comm. and Image Proc.*, vol. 2308, (Chicago, Illinois), pp. 389–400, Sept. 1994.