by certain amount while decreasing the difference between x_2 and $\hat{x}_2$ by the same amount. This means that the MAE criterion can tolerate large errors at certain points since these errors will be averaged to other points, leading to the minimum MAE. Applying this fact to the restoration of image in impulse noise (where impulse noise is not heavy), it is then easy to understand that a minimum MAE stack filter or PSF would leave some noise in the image (corresponding to large absolute errors) but restore the image details well. However, when the impulse noise becomes heavier, one will experience that the optimal filter tends to the median filter. This is because more impulse noise means that more image pixels have been corrupted by the noise so that the filter cannot tolerate too many large errors any more; instead, it will tend to the median filter, which has the best noise-removing capability in the class of stack filters.

REFERENCES

[1] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, Jr., "Stack filters," *IEEE Trans. Acoust. Speech Signal Processing,* vol. ASSP-34, pp. 898–911, Aug. 1986.

[2] E. J. Coyle and J.-H. Lin, "Stack filters and the mean absolute error criterion," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 36, pp. 1244–1254, Aug. 1988.

[3] E. J. Coyle, J.-H. Lin, and M. Gabbouj, "Optimal stack filtering and the estimation and structural approaches to image processing," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 37, pp. 2037–2066, Dec. 1989.

[4] J.-H. Lin, T. M. Sellke, and E. J. Coyle, "Adaptive stack filtering under the mean absolute error criterion," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 38, pp. 938–954, June 1990.

[5] M. Gabbouj and E. J. Coyle, "Minimum mean absolute error stack filtering with structural constraints," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 38, pp. 955–968, June 1990.

[6] B. Zeng, M. Gabbouj, and Y. Neuvo, "A unified design method for rank order, stack, and generalized stack filters based on classical Bayes decision," *IEEE Trans. Circuits Syst.,* vol. 38, pp. 1003–1020, Sept. 1990.

[7] J.-H. Lin and E. J. Coyle, "Minimum mean absolute error estimation over the class of generalized stack filters," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 38, pp. 663–678, Apr. 1990.

[8] J. P. Fitch, E. J. Coyle, and N. C. Gallagher, Jr., "Median filtering by threshold decomposition," *IEEE Trans. Acoust. Speech Signal Processing,* vol. ASSP-32, pp. 1183–1188, Dec. 1984.

[9] S. K. Mitra, M. R. Petraglia, and L. Onural, "Median filtering by a generalized threshold decomposition," in *Proc. Euro. Conf. Circuit Theory Des.* (Brighton, UK), Sept. 1989, pp. 349–353.

[10] B. Zeng, H. Zhou, and Y. Neuvo, "FIR stack hybrid filters," *Opt. Eng. Special Issue Visual Commun. Image Processing III,* July 1991, pp. 965–975.

Efficient Quadtree Coding of Images and Video

Gary J. Sullivan and Richard L. Baker

Abstract— **The quadtree data structure is commonly used in image coding to decompose an image into separate spatial regions to adaptively identify the type of quantizer used in various regions of an image. We describe the theory needed to construct quadtree data structures that optimally allocate rate, given a set of quantizers. A Lagrange multiplier method finds these optimal rate allocations with no monotonicity restrictions.**

We use the theory to derive a new quadtree construction method that uses a stepwise search to find the overall optimal quadtree structure. The search can be driven with either actual measured quantizer performance or ensemble average predicted performance. We apply this theory to the design of a motion compensated interframe video coding system using a quadtree with vector quantization.

I. INTRODUCTION

Image coding is commonly used to reduce the cost of image communication and storage. One problem with low-rate coding is the spatial nonstationarity of most images. Efficient image and video coding algorithms compensate for nonstationarity by adapting spatially, allocating extra rate to highly detailed areas in the image. Shoham and Gersho [1], [2] studied the allocation of rate among a set of quantizers, given some segmentation. We use their analysis to instead find the optimal segmentation of an image given the constraint that this allocation is communicated using a quadtree data structure.

The use of quadtree decomposition for image coding has been considered by several researchers [3]–[8]; however, none have incorporated a method to find truly *optimal* quadtree structures, given some additive distortion measure such as squared error or absolute difference. One method capable of finding such optimal structures was recently described by Chou *et al.* using a generalized version of the Breiman, Friedman, Olshen, and Stone (BFOS) algorithm [9]. It turns out that the theoretical underspinnings of generalized BFOS (G-BFOS) are identical to the rate allocation theory discussed by Shoham and Gersho [1], [2]. In this correspondence, we develop algorithms to build optimal quadtree structures which use the simple Lagrange multiplier concept underlying both methods, and we compare them with G-BFOS. A preliminary description of our work on this topic was presented in [10].

The quadtree data structure decomposes a $2^N \times 2^N$ image block into an $(N - n_0 + 1)$-level hierarchy, where all blocks at level n have size $2^n \times 2^n, 0 \leq n_0 \leq n \leq N$ (see Fig. 1). This structure corresponds to a tree, where each $2^n \times 2^n$ block (called a *node*) can either be a *leaf,* i.e., it is not further subdivided, or can branch into four $2^{n-1} \times 2^{n-1}$ blocks, each a *child* node. The tree can be represented by a series of bits that indicate termination by a leaf with a "0" and branching into child nodes with a "1" (Fig. 1).

Vector quantization (VQ) [9], [11] achieves compression by approximating each vector of source data by a *codeword* selected from a countable *codebook* vector set. A fundamental result of rate-distortion

Manuscript received November 23, 1990; revised July 31, 1992. This work was supported by Apple Computer, Eastman Kodak, and Rockwell International through the University of California MICRO program and by the Hughes Aircraft Doctoral Fellowship Program. This work was orginally performed while the authors were with the University of California, Los Angeles, USA.

The authors are with PictureTel Corporation, Danvers, MA, USA 01923.

IEEE Log Number 9216574.

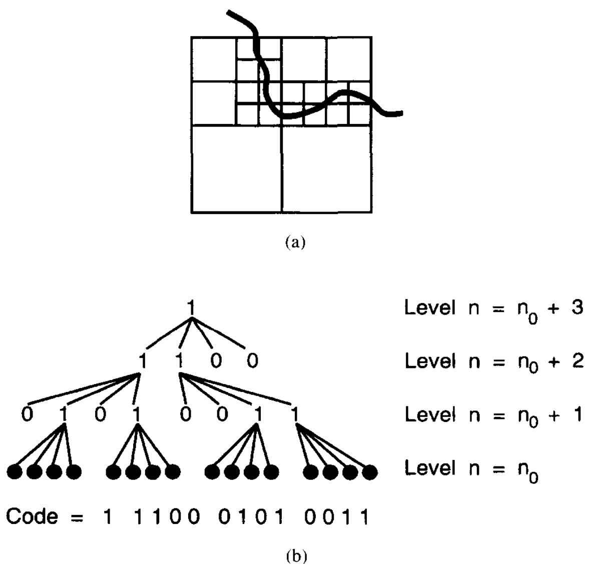

Fig. 1. Quadtree data structure: (a) Image block with signal to be represented; (b) decision bits encoding quadtree structure.

theory states that coding blocks rather than scalars can achieve better performance, even for independent input data [12]. However, we are not concerned here with any particular type of quantizer; rather, we consider any quantizer that associates each quadtree leaf node with an incurred rate and distortion.

In Section II, we present algorithms using Lagrange multiplier rate allocation theory for quadtree construction. Section III describes methods for finding the proper Lagrange multiplier to use with each algorithm. Simulation results are presented in Section IV and conclusions in Section V. We note that while the theory described here is for quadtree systems, it applies equally well to other decomposition structures (e.g., oct-tree, binary-tree, etc.) and to joint optimization of tree structures with additional coding parameters, such as Strobach's joint motion vector optimization [6], [13], [14].

II. Algorithms

Quadtrees are typically constructed by *top-down* or *bottom-up* methods, although other constructions are possible. In top-down construction [3], [8], a judgement is first made as to whether the entire block can be represented by a single leaf or whether it must be divided into four subblocks. If the block is divided, then a binary decision is made for each subblock to determine whether it needs further division, and so on. Bottom-up construction [5]–[7] consists of binary decisions to merge, where construction begins with the smallest possible block size in the quadtree. If all relevant subblocks have been combined into a larger block, then a decision is made whether to combine the larger blocks into a yet larger block, and so on. Any arbitrary (possibly nonsequential) method could likewise determine the desired subdivided structure of an image block.

One need not exhaustively consider all possible rate allocations in order to identify the optimal quadtree in the set of all possible quadtrees. In this section, we present an algorithm that finds the optimal tree and is only slightly more complex than bottom-up construction.

An efficient compression coding algorithm must minimize rate as well as distortion. A choice between quadtree structures amounts to a choice between points in the R-D plane. Using a Lagrange multiplier $\lambda \geq 0$, we can find points on the convex hull of all possible R-D pairs [1], [2] by minimizing the *objective function*

$$\min \{d_k + \lambda b_k\} \tag{1}$$

for each separate region k, where b_k is the number of bits assigned to the region, and d_k is the resulting distortion. We call this the *principle of separate minimization* since objective function minimization for each separate region results in the global optimum in the unconstrained sense described by Shoham and Gersho [1], [2].

Assume we wish to find the optimal quadtree structure for block X_{n+1} of size $2^{n+1} \times 2^{n+1}$ and that we already know optimal (minimum objective function) quadtree structures for its four $2^n \times 2^n$ subblocks $X_{n,i}, i = 1, \ldots, 4$. Denote the total bits and distortion of these optimal constituent subtrees as $b_n^*(X_{n,i})$ and $d_n^*(X_{n,i}), i = 1, \ldots, 4$, and denote the bits and distortion incurred by representing the block with a single leaf at level $n+1$ as $b_{n+1}(X_{n+1})$ and $d_{n+1}(X_{n+1})$. Using (1), the four subtrees should be combined into a leaf whenever

$$\Delta d \leq \lambda \Delta b \tag{2}$$

where

$$\Delta d = d_{n+1}(X_{n+1}) - \sum_{i=1}^{4} d_n^*(X_{n,i}) \tag{3}$$

and

$$\Delta b = \sum_{i=1}^{4} b_n^*(X_{n,i}) - b_{n+1}(X_{n+1}) \tag{4}$$

are the distortion increase and rate savings caused by combining the four subtrees into a single leaf node. Assuming the quadtree structure is coded as in Fig. 1, i.e., one bit per merge decision, the number of bits needed to represent the subtree for block X_{n+1} would be

$$b_{n+1}^*(X_{n+1}) = \begin{cases} 1 + b_{n+1}(X_{n+1}) & \text{if } \Delta d \leq \lambda \Delta b \\ 1 + \sum_{i=1}^{4} b_n^*(X_{n,i}) & \text{otherwise.} \end{cases} \tag{5}$$

The resulting distortion would be

$$d_{n+1}^*(X_{n+1}) = \begin{cases} d_{n+1}(X_{n+1}) & \text{if } \Delta d \leq \lambda \Delta b \\ \sum_{i=1}^{4} d_n^*(X_{n,i}) & \text{otherwise.} \end{cases} \tag{6}$$

This process results in a quadtree having $n - n_0 + 2$ levels, which is optimal, as we now show. If block X_{n+1} is coded as four separate subtrees, the principle of separate minimization (1) requires that for the structure to be optimal, its subcomponents must also be optimal [1], [2]. We need not consider any quadtree structure that contains subtrees which are not optimal. Therefore, only two alternatives need be considered to identify the optimal quadtree having $n - n_0 + 2$ levels: the use of a single combined leaf node or the use of four optimal subtrees having $n - n_0 + 1$ levels.

The optimal quadtree can be found by starting at the $n = n_0$ level. At that level, only trivial one-level (single leaf) subtrees can be used, i.e., $b_{n_0}^*(X_{n_0,i}) = b_{n_0}(X_{n_0,i})$ and $d_{n_0}^*(X_{n_0,i}) = d_{n_0}(X_{n_0,i})$. We next use (2) through (6) to find optimal two-level subtrees. Once they are known, we can find optimal three-level subtrees, and so on, until the optimal $(N - n_0 + 1)$ -level quadtree is known. By moving from one optimal construction to the next, many admissible but suboptimal structures are eliminated from consideration.

Conventional bottom-up and top-down algorithms are also easily derived using localized merge/split decisions based on the objective function criterion (1). A method essentially of this type was recently independently developed by Strobach [7]. However, such an

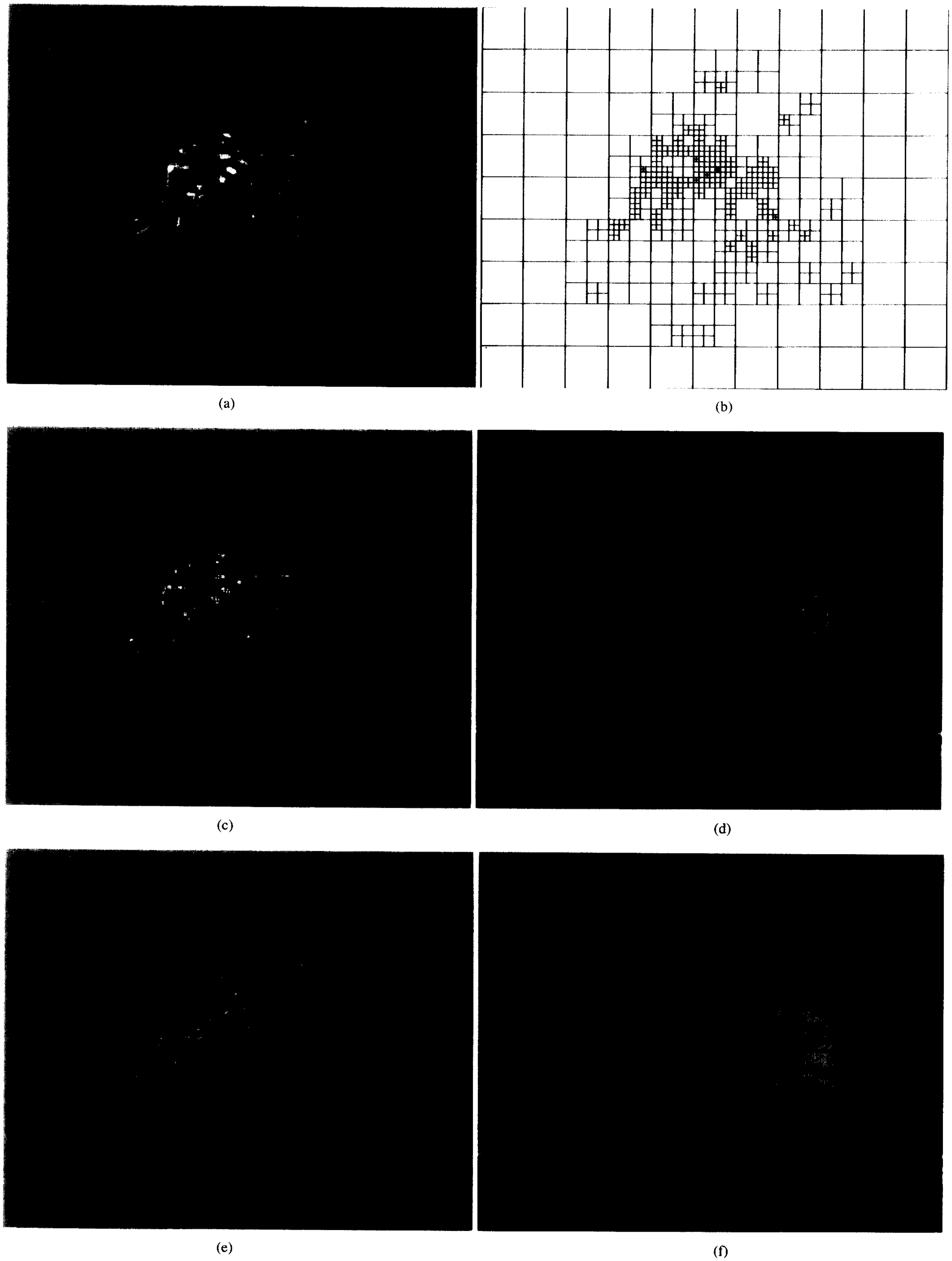

Fig. 2. Frame transition illustration: (a) Prediction error (2 * error +128); (b) quadtree structure; (c) quantized error image; (d) final frame estimate; (e) residual error; (f) original frame.

algorithm is not optimal over all possible trees, as is the algorithm using (2) through (6). Our new method is not a bottom-up algorithm in the conventional sense as it considers merging each set of four subtrees into a single node representation regardless of whether the four subtrees were merged to a single node each in prior steps of the tree construction process.

The derivation above assumes actual rates and distortions are known, yielding an *ultimate bit allocation* (UBA) [1], [2]. In order to obviate the computational complexity involved in directly measuring rate and distortion by VQ encoding, one can use approximate values based on ensemble average allocation (EAA) performance. Some statistic such as local energy, variance, maximum absolute deviation, or a combination of statistics can be measured from the block under consideration and used to predict quantizer performance. The quadtrees can then be constructed by minimizing the expected value (as opposed to the actual value) of the objective function. If local energy is used and each quantizer's distortion is approximated as being linear with input energy, the optimal ensemble method becomes a simple threshold test [14].

III. The Lagrange Multiplier

The Lagrange multiplier λ controls which $D(R)$ point will be chosen among the possible allocations in the R-D plane. If the optimal algorithm is used, the $D(R)$ point will lie on the convex hull of all possible allocations, and rate will be a monotone decreasing function of λ, with distortion monotone increasing [2]. Their precise functional relationships are unknown; therefore, a 1-D search must be performed to determine the appropriate value of λ for a given rate or distortion constraint.

A. Bracketing Interval Search for λ

A straightforward way of conducting this search would be simply to test values of λ, repeatedly building the quadtree allocation until the desired point is located. Fortunately, this process is relatively easy to repeat after it has been perfomed once. For example, for an ensemble average allocation, the statistic at each level need only be measured once. The allocation can then be built over and over using these same measurements. In the case of UBA, the data block must be completely encoded once by each VQ codebook. The resulting rate and distortion measurements can then be stored and the allocation built over and over using these measurements.

In an interframe application, a good initial guess for λ would be the final value used for the previous frame. The initial value of λ can then be adjusted, perhaps by trying some percentage increase or decrease based on the resulting rate. Once upper and lower bounds are obtained, this *bracketing interval* can be successively decreased in size to improve accuracy, perhaps by a bisection search [15].

An alternate method for decreasing the bracketing interval size in the optimal algorithm (when using convex hull points) is to test a "critical" value of λ for which both ends of the interval are equally good. Let $\lambda_l < \lambda_u$ be two values of λ for which the optimal rate $R_l > R_u$ and distortion $D_l < D_u$ are known. The critical value λ_{cr} is given by

$$\lambda_{cr} = \frac{D_u - D_l}{R_l - R_u}.$$

If no quadtree decomposition exists on the convex hull between those for λ_l and λ_u, then the decompositions for λ_l and λ_u will both also be optimal for λ_{cr}, and testing can cease since no solution exists between λ_l and λ_u. Otherwise, a new decomposition is found that is optimal for λ_{cr} and the bracketing interval can be narrowed, with λ_{cr} as the new upper or lower bound.

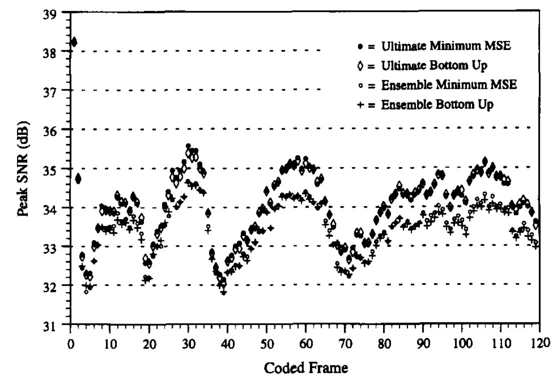

Fig. 3. Peak SNR performance for proposed algorithms.

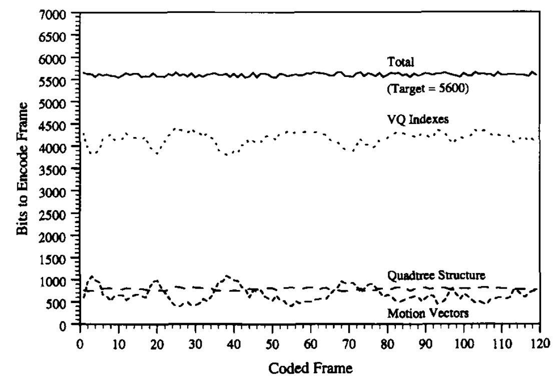

Fig. 4. Rate usage.

B. Directed Search for λ

Other useful information can be obtained from quadtrees corresponding to the end points of a bracketing interval $[\lambda_l, \lambda_u]$. For example, one need not test all possible tree structures to find the quadtree corresponding to some λ, where $\lambda_l < \lambda < \lambda_u$. Rate and distortion are monotonic with λ; therefore, the search region for allocated rates can be narrowed to values between the bracketing rates when using a value of λ that lies between two values having known allocations [2]. This property, with an additional restriction, forms the basis of the G-BFOS algorithm for tree structure optimization [9].

Chou *et al.* have described a method for creating optimal trees and mentioned its applicability to quadtree image coding [9]. Their method assumes that rate is monotone increasing with the depth of the tree structure and that distortion is monotone decreasing. A reduction in the depth of a branch of the tree must always correspond to a decrease in rate in order for their algorithm to work properly. Thus, for $\lambda = 0$, the optimal tree would have all leaf nodes at level n_0, and for $\lambda = \infty$, only one node at level N. Under this assumption, convex hull trees nest, i.e., if $\lambda_l < \lambda_u$, then the optimal tree for λ_u is a subtree of the optimal tree for λ_l.

Since convex hull trees nest under this monotonicity assumption, one can find all points on the convex hull of distortion-rate points by starting with $\lambda = 0$ and "pruning" the tree in an optimal fashion, each time moving from the current value of λ to the next larger value having a different (smaller) tree structure. Chou *et al.* have described an efficient way to identify and prune the appropriate portions of the tree at each step. Their method for determining λ starts with $\lambda = 0$

and proceeds stepwise across *singular* values of λ (values of λ with multiple solutions).

Shoham and Gersho also describe a method using singular Lagrange multipliers for the more general rate allocation problem (not necessarily using a tree structure) and without the simplifying restriction of monotonicity [2]. This theory was recently used to develop a general tree-structure optimization algorithm, which can also jointly optimize the quadtree structure with the VQ index used at each leaf node [13], [14].

Either the bracketing interval iterative approach or a step-wise sequential approach such as G-BFOS can be used to find the optimal tree structure. In either case, the complexity of the λ search is low compared with the complexity of other image coding algorithmic components such as VQ or DCT.

C. Allocations Above the Convex Hull

It is possible that the solution to the allocation problem with a rate constraint lies above the convex hull points that can be found by solving the unconstrained problem [2]. For some applications, these *inaccessible solutions* can be avoided by time sharing between two tree structures [9]. However, for quadtree image decomposition, one structure or another must be chosen since the tree will only be used once. Two useful methods of finding allocations above the convex hull have been suggested [2], [16], though neither is optimal.

Shoham and Gersho give an *allocation performance bound* (APB) on the suboptimality of a solution. Better APB's are not difficult to derive [14]. In our testing, the APB for quadtree image coding has been small (always less than 0.03 dB) when using the nearest point on the convex hull with rate below the constraint. Thus, we do not feel the extra complexity needed to find points above the convex hull is justified in this case.

IV. Simulation Results

Simulations for interframe quadtree VQ with motion compensation were conducted for four allocation algorithms described in Section II: 1) Bottom-up with UBA, 2) R-D optimized with UBA, 3) bottom-up with EAA, and 4) R-D optimized with EAA. The EAA algorithms used an energy statistic for performance prediction.

Test data consisted of 120 monochrome frames of the "salesman" sequence sampled at 10 frames per second, with CIF resolution (352×288 pixels). Each test started with a perfect first frame. The data rate was 5600 bits per frame, or 56 kbits per second (0.0552 bits per pixel per frame).

The coding process for the ensemble bottom-up algorithm is illustrated in Fig. 2 for a typical frame transition (the 65th). The motion prediction error signal (a) is mapped to a quadtree structure (b) and quantized by the VQ codebooks. The quantized error signal (c) is added to the motion compensation prediction to form the final frame estimate (d). The remaining error in the estimate is shown in (e) and the original frame in (f).

The quadtrees had five levels, with $n_0 = 1$ and $N = 5$. The codebooks used for levels one through four were fixed-rate full-search codebooks of 256 vectors each. Levels three and four, which have block sizes greater than four by four, were subsampled by averaging two-by-two and four-by-four blocks, respectively, to reduce the VQ block size to four by four. Each quantized mean was then used to represent all of the pixels from which it was generated. Somewhat better performance could be obtained by using a better dimensionality reduction method such as DCT [3] and a better interpolation method such as optimal nonlinear interpolative VQ [17] or some type of quadtree "tent-pole" interpolation [4], [7]. Level 5 vectors were coded with rate zero, i.e., the "codebook" contained a single 32×32 block of zeros. Motion compensated prediction frames were constructed using a reduced-search 16×16 block-matching algorithm. The value of λ was determined by a bisection search. Further details of the simulations are provided in [14].

Fig. 3 shows the peak-SNR performance for the four tested algorithms, which is defined as $10 \log_{10} (255^2/\sigma^2)$, where σ^2 is mean-square error. The bottom-up and R-D optimized UBA algorithms had virtually identical peak-SNR performance, which was usually 0.5–1 dB better than that of the two EAA algorithms. The perceptual difference between the results of the four algorithms was slight. Fig. 4 shows the rates generated by the algorithm for the bottom-up EAA method, which were typical of the other methods as well. The motion vectors and quadtree structure each used about 15% of the overall rate, and the remainder was used for the VQ indices.

V. Conclusions

We have developed the theory needed for the efficient use of quadtree data structures to allocate rate in image coding systems. We used this theory to develop several specific optimal and near-optimal quadtree construction algorithms. The usefulness of these methods was demonstrated by simulations showing their performance for low bit-rate video coding.

References

[1] Y. Shoham and A. Gersho, "Efficient codeboook allocation for an arbitrary set of vector quantizers," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP),* 1985, pp. 43.7.1–43.7.4.

[2] ——, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 36, pp. 1445–1453, Sept. 1988.

[3] D. J. Vaisey and A. Gersho, "Variable block-size image coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP),* Apr. 1987, pp. 25.1.1–25.1.4.

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

[5] C.-Y. Chiu and R. L. Baker, "Quad-tree product vector quantization of images," in *Proc. SPIE Conf. Advances Image Compression Automat. Target Recogn.,* Mar. 1989, pp. 142–153, vol. 1099.

[6] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Trans. Commun.,* vol. 38, pp. 477–486, Apr. 1990.

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

[8] N. M. Nasrabadi, S. E. Lin, and Y. Feng, "Interframe hierarchical vector quantization," *Opt. Eng.,* vol. 28, pp. 717–725, July 1989.

[9] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. Inform. Theory,* vol. 35, pp. 299–315, Mar. 1989.

[10] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP),* May 1991, pp. 2661–2664.

[11] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.,* vol. 4, pp. 4–29, Apr. 1984.

[12] T. D. Lookabaugh and R. M. Gray, "High-resolution quantization theory and the vector quantizer advantage," *IEEE Trans. Inform. Theory,* vol. 35, pp. 1020–1033, Sept. 1989.

[13] G. J. Sullivan and R. L. Baker, "Rate-distortion optimization for tree-structured source coding with multi-way node decisions," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing (ICASSP),* Mar. 1992, pp. III-393-396.

[14] G. J. Sullivan, "Low-rate coding of moving images using motion compensation, vector quantization, and quadtree decomposition," Ph.D. thesis, Univ. of Calif., Los Angeles, Sept. 1991.

[15] W. K. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge, MA: Cambridge University Press, 1988.

[16] S.-Z. Kiang, R. L. Baker, G. J. Sullivan, and C.-Y. Chiu, "Recursive optimal pruning with applications to tree-structured vector quantizers," *IEEE Trans. Image Processing,* vol. 1, pp. 162–169, Apr. 1992.

[17] A. Gersho, "Optimal nonlinear interpolative vector quantization," *IEEE Trans. Commun.,* vol. 38, pp. 1285–1287, Sept. 1990.