Rate-Distortion Optimal Fast Thresholding with Complete JPEG/MPEG Decoder Compatibility

Kannan Ramchandran and Martin Vetterli

Abstract— **We show a rate-distortion optimal way to threshold or drop the DCT coefficients of the JPEG [1] and MPEG [2] compression standards. Our optimal algorithm uses a fast dynamic programming recursive structure. The primary advantage of our approach lies in its complete compatibility with standard JPEG and MPEG decoders.**

I. Introduction

JPEG [1] and MPEG [2] are popular DCT-based compression standards for still images and video sequences, respectively. Standardization of these compression formats has spurred the wide usage of JPEG (and recently MPEG) decoders. The key to good compression when using these standards *while still being compatible with the standard decoder* lies in determining an optimal thresholding strategy for the DCT coefficients. We are thus interested in retaining that subset of the DCT coefficients for the image or video frame that is the "best" in a rate-distortion (R-D) sense. Thresholding or dropping the less significant DCT coefficients may be desirable in the R-D sense, as it may lead, at a marginal sacrifice of coded quality, to a significant reduction in coding bit rate, due to fewer coefficients needing to be transmitted. This is especially so when deciding the last nonzero coefficient, which is typically followed by an inexpensive "end-of-block" code [1], [2]. Thus, it may especially pay to get rid of compression-hindering sparsely interspersed insignificant coefficients that represent the last nonzero values before the end-of-block. In the case of JPEG, where the standard does not permit variable scaling of the different image blocks, intelligent adaptive thresholding of the coefficients would essentially amount to changing the quantization scales at the block level, without breaking the rules of the game!

In this paper, we formulate an R-D optimal strategy to threshold the quantized DCT coefficients by using a fast recursive dynamic programming (DP) technique. Starting from the "highest quality point" after quantization at a fixed scale (for JPEG [1] or QP-level (for MPEG [2]), one can sweep the entire thresholding R-D curve over a continuous range of target bit rates (or equivalently target-coding qualities) by dropping insignificant coefficients in the image or video frame. Thus, our algorithm could find all points that reside on the convex hull of the thresholding R-D curve. The appeal of the strategy lies in its combination of R-D optimality, speed of operation, and its complete compatibility with standard JPEG and MPEG decoders, which remain blissfully oblivious to the thresholding gymnastics performed by the encoder. A point to note is that our algorithm exploits the monotonic nature of bit rate versus the zero run length count preceding a nonzero coefficient inherent in the Huffman tables of JPEG and MPEG, to specify a fast "pruning" operation in the DP recursion. Thus, the computational complexity of the algorithm becomes low enough for it to be implementable.

The problem of thresholding using an "efficiency" measure was tackled in [3] and was the inspiration for the work described here. The approach of [3] does not attain R-D optimality, as is our goal here. Moreover, in this work, we use a faster dynamic programming formulation than that of [3] and additionally exploit the monotonicity of the bit rate versus zero run-length counts of the JPEG and MPEG codebooks to appreciably reduce the computational complexity of the algorithm. The subject of finding a locally optimal quantization matrix that matched the image was tackled in [7], using a computationally intensive descent algorithm which is not block-adaptive.

This paper is organized as follows. Section II defines the problem quantitatively. Section III describes the optimal solution, which is presented in algorithmic detail in Section IV. Finally, coding applications using JPEG and MPEG are described in Section V.

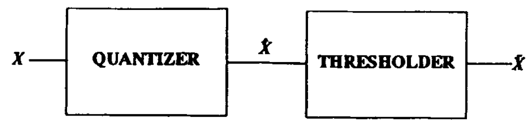

Fig. 1. Block diagram of problem statement: X is the input image, $\hat{X}$ is the quantized prethresholded version (after JPEG or MPEG), and $\tilde{X}$ is the output of the thresholder. We want to minimize the distortion, for a fixed quantizer, between the original and the thresholded images subject to a bit budget constraint.

II. Problem Statement

We wish to find that optimal set of quantized DCT coefficients to be retained for every 8×8 block of an image or video frame such that the mean-squared-error (MSE) distortion (any additive distortion metric is feasible in general, e.g., activity-weighted MSE) between the original image and the thresholded version is minimized *subject to a maximum target coding-bit-rate constraint,* or equivalently, the coding bit rate is minimized subject to a maximum allowable distortion constraint.

Thus, if X is the input signal, $\hat{X}$ the quantized output corresponding to a fixed scale or "anchor" level representing the maximum quality operating point, and $\tilde{X}$ the thresholded version of $\hat{X}$, we seek to minimize the MSE distortion between X and $\tilde{X}$ given the quantized image $\hat{X}$, subject to a total coding-bit budget of R_{budget} for $\tilde{X}$. That is, our goal is to find

$$\min[\mathcal{D}(X, \tilde{X})|\hat{X}] \text{ subject to } \mathcal{R}(\tilde{X}) \leq R_{\text{budget}} \tag{1}$$

where $\tilde{X}$ is a thresholded version of $\hat{X}$ (see Fig. 1).

III. Optimal Solution

A. R-D Optimality: The Constant Slope Condition

The "hard" constrained thresholding problem of (1) can be solved by being converted to an "easy" equivalent unconstrained problem by "merging" rate and distortion through the Lagrange multiplier λ [4]. The unconstrained thresholding problem becomes the determination (for a fixed λ) of that set of coefficients, which results in the minimum total Lagrangian cost defined as

$$J(\lambda) = D(X, \tilde{X}) + \lambda R(\tilde{X}). \tag{2}$$

Manuscript received October 30, 1992; revised August 4, 1993. The work of K. Ramchandran was supported by the New York State Science and Technology Foundation's CAT. The work of M. Vetterli was supported in part by the National Science Foundation under grants ECD-88-11111 and MIP-90-14189.

K. Ramchandran is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801.

M. Vetterli is with the Department of Electrical Engineering and Center for Telecommunications Research, Columbia University, New York, NY 10027-6699.

IEEE Log Number 9402241.

The optimal coefficient search for the image *can be done independently* for every 8×8 image block for the fixed quality "slope" λ, which trades off distortion for rate. This is because it can be shown [4], [5] that, at R-D optimality, all blocks *must operate at a constant slope point* λ *on their R-D curves.*

This result is fairly intuitive as seen from the argument that, at optimality, each allocated bit must do an equal amount of "good" (in the distortion-removal sense), for otherwise one could redistribute bits from "unprofitable" to "profitable" blocks. It is obviously cost-effective to keep doing this until an optimum is reached, where no block can spend the next available bit any more profitably than any other, i.e., until all signal blocks have the same slope (λ) on their rate-distortion function. The desired optimal constant slope value λ^* is not known *a priori* and depends on the particular target budget or quality constraint. Fortunately, however, λ^* can be obtained relatively painlessly via a fast convex recursion in λ using the bisection algorithm [5]. The main advantage of the Lagrangian approach is its independent optimization property for each signal element, enabling the independent analysis of the 8×8 image blocks in our case!

For a more mathematical formulation, if T is $\{0, 1, 2, \cdots, 63\}$, the set of all 8×8 coefficients in each block ordered in the 1-D zigzag scan order, $S \preceq T$ any feasible ordered subset of T, and $D(S), R(S)$ the distortion and bit rate, respectively, associated with retaining the coefficients in S, our problem of finding:

$$D_{\min} = \min_{S \preceq T} D(S) \text{ subject to } R(S) \leq R_{\text{budget}} \qquad (3)$$

is solved by introducing $J(\lambda) = [D(S) + \lambda R(S)]$ representing the Lagrangian cost of S associated with the quality factor λ and solving the following equivalent unconstrained problem

$$J_{\min}(\lambda) = \min_{S} J(\lambda) = \min_{S}[D(S) + \lambda R(S)]. \qquad (4)$$

The desired optimal constant slope value λ^* is not known *a priori* and depends on the particular target budget or quality constraint but is obtained using a fast convex search using the bisection algorithm [5]

$$J_{\min}(\lambda^*) = \max_{\lambda \geq 0}[J_{\min}(\lambda) - \lambda R_{\text{budget}}] \qquad (5)$$

B. Fast Dynamic Programming Algorithm

Since the optimal convex-hull solution can be found, as described above, by independently finding the minimum-Lagrangian-cost operating point (i.e., one that minimizes $J = D_{\text{block}} + \lambda^* R_{\text{block}}$) for each block of the sequence, it suffices to consider a single block for analysis. The problem is solved using a dynamic programming approach.

The zigzag scan that is part of the standards [1], [2] is used to order the 2-D coefficients. As an initialization, one has to gather the $\Delta J_{j,k}$'s associated with the incremental Lagrangian cost of going from coefficient j to coefficient k (i.e., dropping all the coefficients between them) for all nonzero valued (j, k) coefficient pairs with $j < k$. $\Delta J_{j,k} = -E_k + \lambda R_{j,k}$ represents the "net gain" of including k conditioned on the previous nonthresholded coefficient being j. E_k represents the "goodness" measure as calculated by the decrease in squared error caused by not thresholding k, and is given by $C_k^2 - (C_k - \hat{C}_k^2)$, where C_k and $\hat{C}_k$ are the unquantized and quantized coefficient values, respectively, while R_{jk} is the conditional bit rate in encoding coefficient k, given that the previous nonzero coefficient is j, i.e., it is the conditional cost of not thresholding k. See Fig. 2. the values R_{jk} can be "read off" from the standard Huffman coding tables for both JPEG and MPEG and can be prestored. Note that to find the optimal algorithm, only the run lengths need to be stored, and not the actual Huffman-coded bit stream, and this represents a trivial memory requirement.

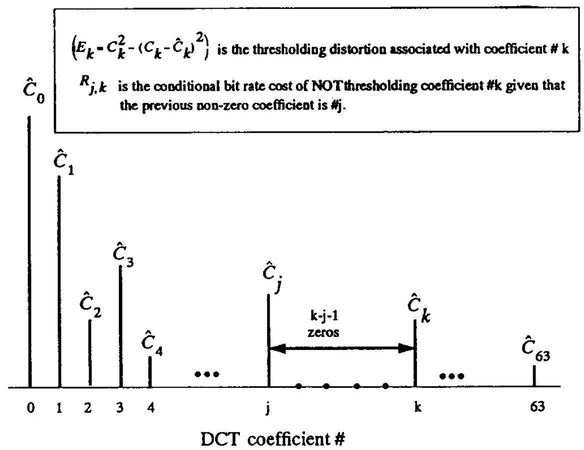

Fig. 2. DCT coefficients of a typical 8×8 image block of JPEG/MPEG ordered in 1-D according to the zigzag scan order. Coefficient #0 is the dc coefficient. E_k and R_{jk} are shown as the thresholding distortion and nonthresholding cost associated with coefficient k conditioned on the previous nonzero coefficient being j.

The optimal operating point, for a fixed value of λ, can be done (see Fig. 3) in a recursive fashion by finding (i) the minimum Lagrangian cost, $J^*(k)$, and (ii) the optimal predecessor coefficient, predecessor (k), associated with choosing coefficient k as the last nonzero coefficient for all $k = 1, 2, \cdots, 63$. Then, starting from that coefficient k^*, which is the cheapest to retain as the last nonzero coefficient, i.e., minimum $J^*(\cdot)$, the optimal set can be "backtracked" from the optimal predecessor chain calculated for all predecessors of k^*.

A more elaborate step-by-step description of the algorithm follows. The recursion begins with coefficient 0. The cost of dropping all ac coefficients is stored in $J^*(0)$. Then, one proceeds with the minimum cost "path" that ends in coefficient 1. There is not much choice here, as there is only one path that ends in coefficient 1, namely, dropping all coefficients from 2 to 63. This cost is saved in $J^*(1)$, and the optimal predecessor to 1 is obviously 0. Proceeding to coefficient 2, the most efficient recursive way of determining the best path that ends in 2 is to find the *optimal predecessor* to 2, i.e., either 0 or 1. Since the optimal costs associated with ending at 0 and 1 are known from $J^*(0)$ and $J^*(1)$, respectively, the job of finding the cheapest cost path ending in 2 is simply the minimum of $[J^*(0) + \Delta J_{02}]$ (where ΔJ_{02} is the incremental cost of going from 0 to 2) and $[J^*(1) + \Delta J_{12}]$. The smaller of these two costs is saved in $J^*(2)$, and the optimal predecessor of 2 (i.e., the one among 0 or 1 responsible for the smaller total cost leading to 2) is saved in predecessor (2). Proceeding similarly to coefficient 3, the best path ending in 3 has to have a direct predecessor that is either 0 or 1 or 2. As the best costs associated with *ending at all predecessors* are known from previous iterations and are stored in J^* (predecessor), and the incremental cost of going from each predecessor to 3 is known from the precomputed $\Delta J_{\text{predecessor},3}$ for all predecessors 0, 1, and 2, the best path ending in 3 is computed as the cheapest of the total costs $[J^*$ (predecessor) + $\Delta J_{\text{preecessor},3}$ for all predecessors 0, 1, and 2. This cheapest cost is saved in $J^*(3)$, the optimal predecessor is saved in predecessor (3), and the recursion continues to coefficient 4 and so on until the last coefficient 63 is done. At this point, the optimal last nonzero coefficient k^* is obviously the one with the smallest $J^*(k)$ for $k = 0, 1, \cdots, 63$. See Fig. 3. By backtracking from k^*, one can

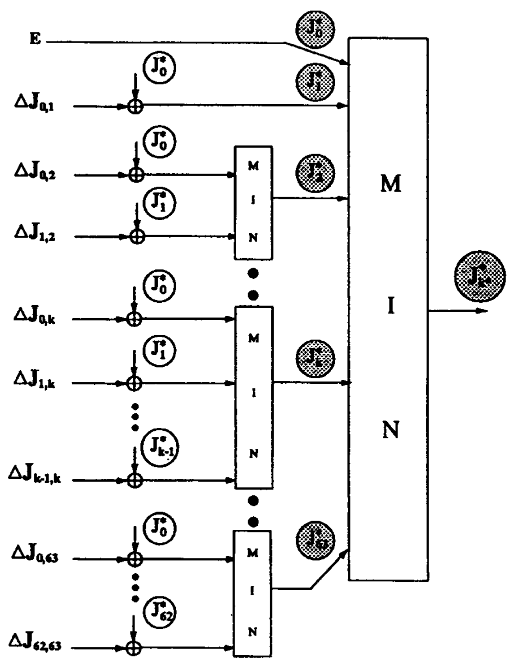

Fig. 3. Block diagram of the dynamic programming recursion of the optimal algorithm for each 8×8 image block. E is the ac energy, $\Delta J_{i,j}$ is the incremental Lagrangian cost of going from coefficient i to coefficient j while dropping all coefficients inbetween, and J_k^* is the minimum Lagrangian cost associated with ending at coefficient k. Note that $k^* = \min^{-1}_{0 \le k \le 63} J_k^*$ is the optimal "last" coefficient.

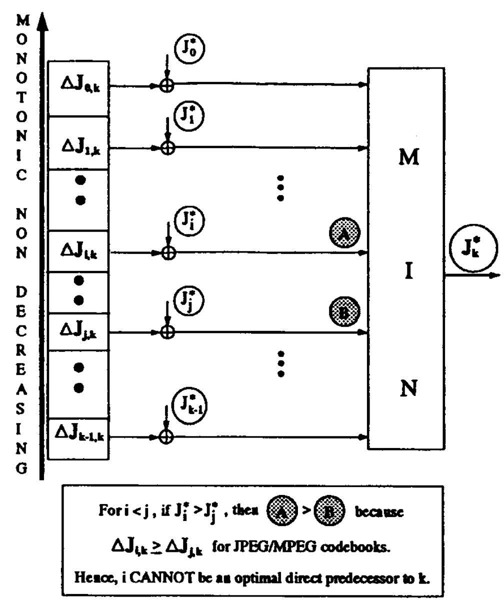

Fig. 4. Fast pruning of nonoptimal predecessors to coefficient k made possible thanks to the monotonicity of the JPEG/MPEG codebooks in the runlengths preceding a nonzero coefficient.

find the optimal predecessor chain sequence starting from predecessor (k^*) and going back to 0, at which point the entire optimal set of coefficients to be retained for each block is known for the given λ.

In finding the optimal predecessor at a particular iteration k as described above, one generally has to consider as candidates *all* coefficients $j < k$. However, for the particular case of monotonicity of R_{jk} in the zero run length count $(k - j - 1)$ (see Fig. 4), which is true for the default coding tables of JPEG and MPEG,[1] a fast pruning algorithm can be used to speed up the search. This results in a substantial decrease in computational complexity and leads to a fast optimal algorithm. The above optimal dynamic programming algorithm is performed independently on all blocks. The composite R-D point for the picked λ is obtained simply as the sum of the optimally obtained R-D points for each block for that λ. Finally, the optimal slope λ^* that solves the desired budget or quality constraint is found using a fast convex search.

IV. Algorithm

We now explain somewhat rigorously the algorithm employed to find the optimal solution to our problem. The optimal algorithm flowchart for a fixed operating slope λ will be described for a single typical 8×8 image block in Phase I. Note that the algorithm is applied independently and in parallel to each signal block to determine the optimal coefficient sequence to be retained for that block. Then in Phase II, the optimal operating slope λ^* for the composite problem will be obtained.

[1] This is a highly reasonable condition even if custom codebooks are used.

A. Data Gathering

Prior to running the optimal algorithm, a one-time fixed cost of gathering the statistics needed for running the optimal algorithm must be endured. See Fig. 2. This involves gathering, for each DCT coefficient k, its thresholding distortion E_k and its conditional nonthresholding coding cost R_{jk} conditioned on every preceding nonzero coefficient $j < k$.

B. Phase I: Finding the Optimal Set of Coefficients for a Given λ

Note that, in the algorithm flowchart to be described, E refers to the total unquantized ac energy in the signal block; i.e., $E = \sum_{k=1}^{63} C_k^2, E_k$ refers to the thresholding distortion associated with coefficient k, R_{jk} refers to the incremental bit-rate cost of coding k after $j, \Delta J_{jk}$ refers to the incremental Lagrangian cost of including k after j, J_k^* is the minimum Lagrangian cost associated with having k as the last nonzero coefficient, and S_k is the set of all candidate optimal predecessor coefficients to k. See Fig. 3.

1) Finding the Optimal LAST Coefficient:

1) For the λ of the current iteration, compute $\Delta J_{ij} = -E_j + \lambda R_{ij}$ for all nonzero coefficient pairs i, j with $j > i$.
2) (Initialization) $k^* \leftarrow 0$; $k \leftarrow 0$; $S_0^* \leftarrow \{0\}$; $J_0^* \leftarrow E$; predessor(0) $\leftarrow$ nil.
3) $k \leftarrow k + 1$; If $k = 64$, go to *Step 7.* Else, continue to the next step.
4) If $E_k = 0$, set $S_k \leftarrow S_{k-1}$ and go to *Step 3.* Else, continue to the next step.
5) $J_k^* \leftarrow \min_{i \in S_{k-1}}[J_i^* + \Delta J_{ik}]$. If $J_k^* \le J_{k^*}^*, k* \leftarrow k$.
6) $S_k \leftarrow \{k\} \cup \{i | (i \in S_{k-1}$ and $J_i^* < J_k^*)\}$.
predecessor(k) $\leftarrow \min^{-1}_{i \in S_{k-1}}[J_i^* + \Delta J_{ik}]$. Go to *Step 3* .

The best "path" ends in k^*, i.e., the optimal set of coefficients to be retained for the given λ for the current block has coefficient k^* as its LAST nonzero coefficient.

2) Backtracking to Find Entire Optimal Set of Coefficients:

1) Initialize the set of optimal coefficients as optset $\leftarrow \{k^*\}$.
2) If predecessor $(k) =$ nil, go to *Step 10.* Else continue to the next step.
3) Get the optimal predecessor to k and include its membership in the set $\{\text{optset}\}$. $\{optset\} \leftarrow \{optset\} \cup \{\text{predecessor}(k)\}$. Go to *Step 8.*
4) Done! Optimal solution of coefficients to be retained for given λ is the set $\{\text{optset}\}$.

Note that a key operation that ensures a fast algorithm is the pruning action described in *Step 6.* This step (see Fig. 4) eliminates from contention for predecessor to the next nonzero coefficient, all prior coefficients whose lowest cost of retaining as the last nonzero coefficient exceeds that of the current iteration's coefficient. Thus, if the current coefficient produces the lowest cost so far, *it is the only candidate* for predecessor to the next nonzero coefficient! This is due to the monotonic nature of the bit rate versus zero run length Huffman tables for JPEG and MPEG, where the cost of coding a nonzero coefficient is monotonically nondecreasing in the length of the zero-run preceding that coefficient.

V. Application

The optimal algorithm outlined in Section III can be used to quantify the benefits of adaptive thresholding applied to the JPEG and MPEG coding environments. R-D curves are obtained by sweeping the Lagrange multiplier λ through all positive values for typical quantization scales of interest. Fig. 5 shows the R-D curves for a typical image ("House") using JPEG for prethresholding quantization scales of 1.0 and 0.7. Note the significance of Fig. 5. Point X on curve "a" is the unthresholded "reference" obtained for a scale of 1.0. Let us fix the bit rate for the problem at the reference X's bit rate of 0.615 b/pixel. Now, to see the advantage of optimal thresholding, observe curve "b" corresponding to a finer quantization scale of $s = 0.7$ instead of 1.0. For this finer scale, the nonthresholded bit rate, corresponding to point Z, is obviously greater than that of X. However, if we now start thresholding optimally until the bit budget constraint imposed by curve "a" is satisfied, we will get an adaptive thresholding gain in terms of increased SNR for the same bit rate. Thus, point Y enjoys a 0.7 dB gain at the same bit rate over X. Alternatively, if we fix the PSNR according to that of X (35.7 dB), point W enjoys a compression advantage at the same PSNR of roughly 15%. Fig. 6 shows a similar curve using optimal thresholding using an MPEG intraframe codebook, applied to an intraframe coded frame of the "mit" sequence.

In our experiments, we found that "backing off" to a finer quantization scale and thresholding optimally until we achieved the same reference bit rate (or PSNR) as an unthresholded coarser quantized version resulted in a decent coding gain. However, there was an optimal back-off point, beyond which the performance started to degrade. See Fig. 6. Intuitively, overdoing the act of thresholding after starting with a finer quantization scale is inadvisable beyond a point, as the gain of representing the nonthresholded coefficients with less distortion is no longer outweighed by the drastic step of dropping entire coefficients, since for fine quantization scales, there is not much distortion to begin with.

While the "optimal back-off point" depends on he particular input image as well as the target bit budget (or quality) constraints, it was found in all our simulations (corresponding to typical images and video sequences used in the image-processing community) that the thresholding gain is a convex function of the amount of "back-off" from the reference point. Thus, not "backing off" enough is suboptimal, as is "backing off" too much, with the best performance achieved somewhere (uniquely) in between. Due to the convex nature of this relationship, binary-search methods can be used to find this optimal back-off point.

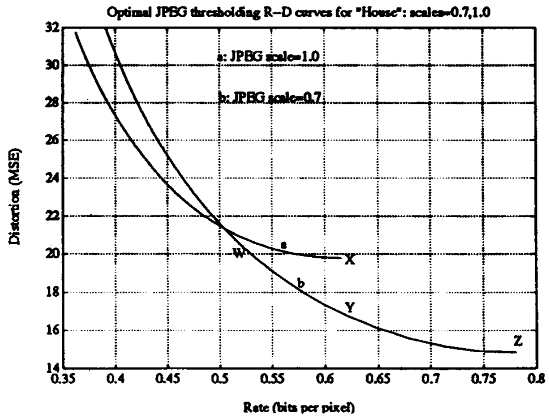

Fig. 5. Optimal thresholding R-D curves for "House" image using JPEG. Curve "a" corresponds to a JPEG quantization scale of 1.0, while curve "b" corresponds to a finer scale of 0.7. Note that if we fix the reference at point X on curve "a" corresponding to a scale of 1.0, we can achieve point Y by "backing off" to a finer scale of 0.7 (point Z on curve "b") and then thresholding optimally to point Y at the same bit rate as X. Note that the thresholding gain for this example (reference X) is approximately 0.7 dB at a bit rate of 0.615 bpp (point Y), or alternatively about 15% reduction in bit rate (point W of curve "b") at the same PSNR of 35.17 dB.

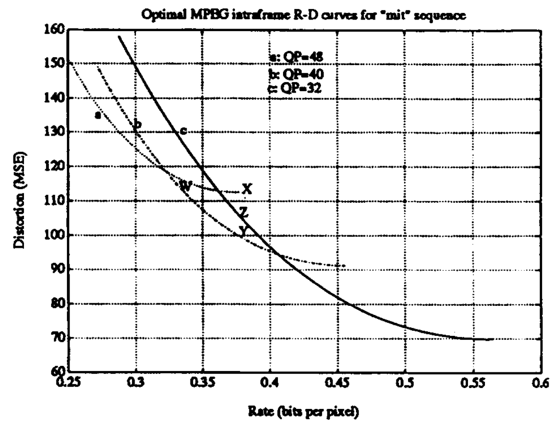

Fig. 6. Optimal thresholding R-D curves for an intraframe coded frame of the "mit" sequence using MPEG. Curve "a" corresponds to a QP level of 48, curve "b" corresponds to a finer quantizer QP level of 40, and curve "c" corresponds to a still finer QP = 32. Note that if we fix the reference at point X on curve "a" corresponding to a QP of 48, we can achieve point Y by "backing off" to the finer QP = 40 and thresholding optimally to point Y at the same bit rate as X. Note that the thresholding gain for this example (reference X) is approximately 0.52 dB at a bit rate of 0.377 bpp (point Y), or alternatively about 12% reduction in bit rate (point W of curve "b") at the same MSE of 112.5. Note that if we back off too much to curve "c" with QP = 32, the achieved coding gain diminishes to point Z (about 0.26 dB) over point X.

Coding results obtained from performing optimal thresholding on typical images and video-sequence frames used in the image-processing community revealed a coding gain of about 0.5–1 dB or alternatively about 12–15% bit-rate compression improvement

(a) (b)

Fig. 7. Subjective results of optimal thresholding in a JPEG framework for the "House" image: (a) Unthresholded image (JPEG scale = 3.0, PSNR = 31.5 dB, bit rate = 0.28 bits per pixel); (b) optimally thresholded image (JPEG scale = 2.0, PSNR = 32.3 dB, bit rate = 0.28 bits per pixel).

while retaining complete decoder compatibility. Optimal thresholding seems to be most subjectively beneficial in the case of low to medium bit-rate coding, as evidenced by Fig. 7. Fig. 7(a) shows as a nonthresholded reference the "House" image coded with JPEG using a quantization scale of 3.0. The thresholded versions using a scale of 2.0 are shown in Fig. 7(b). The coding gain is 0.8 dB, and as can be seen, the subjective quality is also better. Intuitively, this is because for low-bit-rate applications, it is better to represent the low-frequency coefficients with maximum fidelity while dropping the expensive high-frequency coefficients. This gives a smoother but less noisy image, which is the best one can do at low bit rates. Thus, adaptive thresholding can take the place of noise shaping or low-pass filtering without any external processing and without the decoder "skipping a beat."

Using the optimal algorithm as a benchmark, we compared the performance of a fast heuristic algorithm that retains the K largest (in absolute magnitude) DCT coefficients for each 8×8 block for $K < 64$ as in [6]. When the heuristic of [6] is extended to a JPEG quantization framework, it can be seen from Fig. 8 that considerable gain can be obtained (over 5 dB) by resorting to our optimal algorithm, with the gain getting larger as the number of retained coefficients K per block decreases. Our optimal thresholding algorithm took a CPU time of about 1.3 s on a SPARC-II workstation to run a typical iteration of the thresholding algorithm on a 256×256 image and about 6.1 s on a 512×512 image.

Thus we have established, both subjectively and objectively, the benefits of optimally thresholding the DCT coefficients in a JPEG or MPEG coding environment. The primary advantage of our approach is that it is completely compatible with existing JPEG and MPEG decoders.

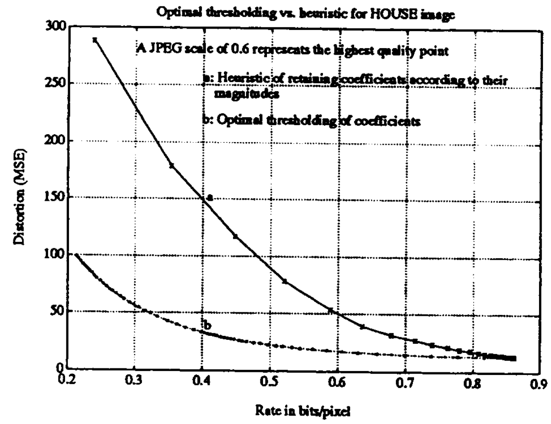

Fig. 8. Comparison in performance of optimal thresholding (curve "b") versus "dumb" heuristic (curve "a") of retaining the largest k coefficients of each block for $k = 1, 2, \cdots, 64$. The highest quality "pivot" point for both schemes corresponds to an unthresholded JPEG scale of 0.6. Note how the optimal scheme beats the heuristic by over 5 dB for bit rates below 0.6 bits per pixel (at 0.52 bits per pixel, the gain is 5.6 dB).

References

[1] "JPEG technical specification: Revision (DRAFT), joint photographic experts group, ISO/IEC JTC1/SC2/WG8, CCITT SGVIII," Aug. 1990.

[2] "MPEG video simulation model three, ISO, coded representation of picture and audio information," 1990.

[3] Y. Yu and D. Anastassiou, "Optimal thresholding for MPEG-based video coding," Tech. Rep. Elec. Eng. Dept., Center for Telecommun., Columbia University, New York, NY, 1991.

[4] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.

[5] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," to be published.

[6] C.-T. Chen and D. LeGall, "A K-th order adaptive transform coding algorithm for image data compression," in *Proc. SPIE: Applications of Digital Image Processing XII*, vol. 1153, 1989, pp. 7–18.

[7] S.-W. Wu and A. Gersho, "Rate-constrained picture-adaptive quantization for JPEG baseline coders," in *Proc. ICASSP-93* (Minneapolis, MN), Apr. 1993.