# Bit Allocation for Dependent Quantization with Applications to Multiresolution and MPEG Video Coders

Kannan Ramchandran, Antonio Ortega, *Student Member, IEEE,* and Martin Vetterli, *Senior Member, IEEE*

***Abstract*— We address the problem of efficient bit allocation in a dependent coding environment. While optimal bit allocation for independently coded signal blocks has been studied in the literature, we extend these techniques to the more general temporally and spatially dependent coding scenarios. Of particular interest are the topical MPEG video coder and multiresolution coders. Our approach uses an operational rate-distortion (R-D) framework for arbitrary quantizer sets. We show how a certain monotonicity property of the dependent R-D curves can be exploited in formulating fast ways to obtain optimal and near-optimal solutions. We illustrate the application of this property in specifying intelligent pruning conditions to eliminate suboptimal operating points for the MPEG allocation problem, for which we also point out fast nearly-optimal heuristics. Additionally, we formulate an efficient allocation strategy for multiresolution coders, using the spatial pyramid coder as an example. We then extend this analysis to a spatio-temporal 3-D pyramidal coding scheme. We tackle the compatibility problem of optimizing full-resolution quality while simultaneously catering to subresolution bit rate or quality constraints. We show how to obtain fast solutions that provide nearly optimal (typically within 0.3 dB) full resolution quality while providing much better performance for the subresolution layer (typically 2-3 dB better than the full-resolution optimal solution).**

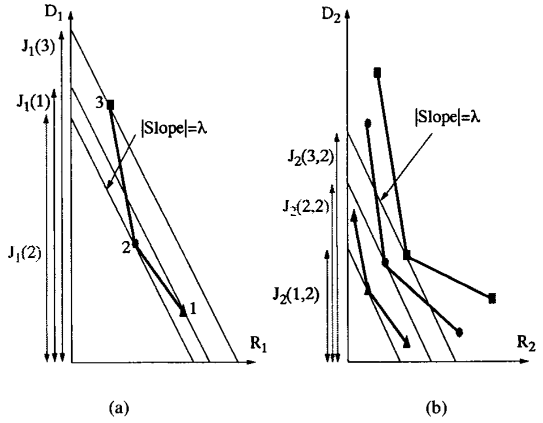


Fig. 1. Operational R-D characteristics of two frames in a dependent coding framework, where frame 2 depends on frame 1: (a) Independent frame's R-D curve; (b) dependent frame's R-D curve. Note how each quantizer choice for frame 1 leads to a different $R_2$–$D_2$ curve. The Lagrangian costs shown are $J = D + \lambda R$ for each frame.

## I. Introduction

The problem of bit allocation, where a given bit budget must be distributed efficiently among a set of given admissible quantization choices, is a classical problem in source coding which has been treated exhaustively in the literature [1]–[4]. A classical framework for bit allocation is rate-distortion theory, which deals with the minimization of source distortion for a given coding bit budget.

In the literature, two approaches have been used to deal with the bit allocation problem: an analytical model-based (continuous) framework and an operational rate-distortion based (discrete) framework. The model-based approach assumes various input distributions and quantizer characteristics from Gaussian inputs [1] to negative-exponential distributed quantizers [2], [3]. With this approach, one can get closed-form solutions based on the assumed models, using continuous optimization theory. Alternatively, since practical coding environments typically resort to a finite set of admissible quantizers, bit allocation for completely arbitrary inputs and discrete quantizer sets has also been addressed in [4] in an operational rate-distortion framework, as defined by the coder and the discrete quantizer choices (see Fig. 1(a)). These quantizers are used by the allocation algorithm to determine the best strategy to minimize the overall coding distortion subject to a total bit budget constraint. This approach uses an integer programming framework to find the optimal discrete solution. In our work here, we resort to this operational rate-distortion (R-D) framework.

The bit allocation problem is specific to the coding environment used. Typical problems include allocating bits optimally among blocks in a DCT-coded still image, or among frames in a video sequence, or among the bands of a subband or wavelet coder, or among the layers of a pyramid coder. All the work addressed in the literature so far has been confined to coding environments where the input signal units (e.g., image blocks or subbands) have been coded *independently*. However, many popular schemes (e.g., DPCM) involve *dependent* coding frameworks, i.e., where the set of available R-D operating

Manuscript received March 10, 1994; revised April 6, 1994. This work was supported by the New York State Science and Technology Foundation's CAT, the Fulbright Commission, the Ministry of Education of Spain, and 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 USA.

A. Ortega is with the Department of Electrical and Computer Engineering and Center for Telecommunications Research, Columbia University, New York, NY 10027-6699 USA. M. Vetterli is with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA USA.

IEEE Log Number 9402885.

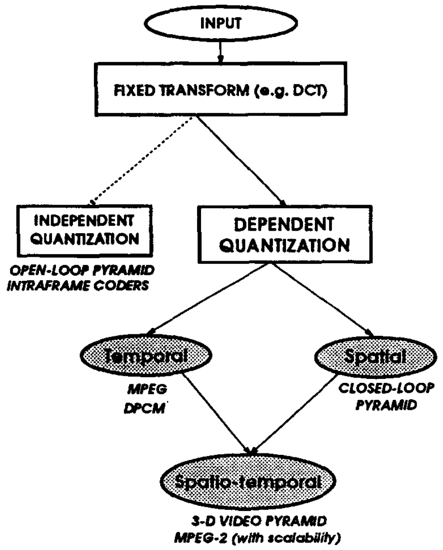


Fig. 2. Overview of typical source coding environments.

points for some coding units (Fig. 1(b)) depends on the particular choice of R-D point for other coding units (Fig. 1(a)). Other dependent coding examples include multiresolution (MR) coders like the Laplacian spatial pyramid [5] and the closed-loop spatio-temporal pyramid video coder [6], as well as the MPEG [7] coder. Note that MR coders are topical due to the current interest in compatible coding schemes as well as their efficiency for digital broadcast of HDTV [8]. Refer to Fig. 2 for an overview of where dependent coding schemes fit into typical source coding environments.

In this paper, extending the work of [9] and [10], we generalize the bit allocation problem to include *dependent* coding units (note that in this work, our signal units can be image blocks or video frames). As in [4], we make no assumptions about the input or quantizer characteristics, and deal with arbitrary input signals and arbitrary discrete quantizer sets. We tackle, under a single framework, coders with temporal and spatial dependencies, such as MPEG and pyramid coders, respectively. We also extend the results of the image pyramid coder to a spatio-temporal video pyramid coding scheme. As seen from Fig. 1 for the two-frame case, the dependent coding problem grows exponentially in the dependency tree depth, which makes the problem very difficult (e.g., DPCM). However, when the dependency tree is structured, as in MPEG, with several "leaves" or "terminal nodes" (e.g., $B$-frames of MPEG, as will be seen in Section III), then it is possible to solve the difficult dependent problem elegantly, by describing how to formulate intelligent pruning rules to eliminate suboptimal operating points.

This paper is organized as follows: in Section II, we provide a Lagrangian-based solution for an arbitrary set of quantizers for each coding block. We point out the complexity of this approach, and introduce a certain monotonicity property of the operational R-D curves of the signal blocks to help reduce the complexity of the search for the optimal solution. We address the temporally dependent coding scheme in detail in Section III, using MPEG as an example, and show how the monotonicity property introduced earlier can be used to formulate fast heuristic solutions. In Section IV, we examine the multiresolution coding scenario, using the Laplacian pyramid as a vehicle, and point out the tradeoffs in achieving full-resolution and subresolution targets, a key question for MR coders. Finally, we tackle the problem of efficient bit allocation for the spatio-temporal video pyramid introduced in [6].

## II. Dependent Coding Problem Formulation

In this section, we define a general dependent allocation problem, show how this general formulation is applicable to MPEG and multiresolution coders, and give a solution based on Lagrange multipliers. Before we introduce the dependent coding problem, let us review the optimal independent allocation case which has been studied in the literature [2], [4].

### *A. Optimal Independent Allocation—The Constant Slope Condition*

The classical rate-distortion optimal bit allocation problem consists of minimizing the average distortion $D$ of a collection of signal elements or blocks subject to a total bit rate constraint $R_{budget}$ for all blocks. For the two-block case, where $\{Q_1, D_1(Q_1), R_1(Q_1)\}$ and $\{Q_2, D_2(Q_2), R_2(Q_2)\}$ refer to the quantizer, distortion and bit rate of each block respectively, the independent allocation problem is

$$\min_{Q_1,Q_2} [D_1(Q_1) + D_2(Q_2)] \tag{1}$$

$$\text{such that} R_1(Q_1) + R_2(Q_2) \leq R_{budget}. \tag{2}$$

The "hard" constrained optimization problem of (1) and (2) can be solved by being converted to an "easy" equivalent unconstrained problem. This is done by "merging" rate and distortion through the Lagrange multiplier $\lambda \geq 0$ [4], and finding the minimum Lagrangian cost $J_i(\lambda) = \min_{Q_i}[D_i(Q_i) + \lambda R_i(Q_i)]$ for $i = 1, 2$. The search for the optimal R-D operating points for each signal block can be done *independently*, for the fixed quality "slope" $\lambda$ (which trades off distortion for rate) because at R-D optimality, all blocks *must operate at a constant slope point* $\lambda$ *on their R-D curves* [4], [11]. The desired optimal constant slope value $\lambda^*$ is not known *a priori* and depends on the particular target budget or quality constraint, but can be obtained via a fast convex search [11].

### *B. General Formulation*

In the more general case, signal blocks may not be independently coded. Without loss of generality, we first consider a two-layer dependency as in Fig. 1. Shown are the R-D characteristics for a given discrete set of quantization choices for the first independent frame $(R_1(Q_1), D_1(Q_1))$ and the second dependent frame $(R_2(Q_1, Q_2), D_2(Q_1, Q_2))$. Our constrained optimization problem is: what quantization choice do we use for each frame such that the total (or average) distortion is minimized subject to a maximum total bit budget constraint? We model the total distortion as a weighted average of the individual distortions $D_1$ and $D_2$ in our general formulation. We will show how different choices of the weights lead to different problems of interest. Our problem can be formulated as

$$\min_{Q_1,Q_2} [w_1 D_1(Q_1) + w_2 D_2(Q_1, Q_2)] \tag{3}$$

$$\text{suchthat} R_1(Q_1) + R_2(Q_1, Q_2) \leq R_{budget}. \tag{4}$$

Note that although $Q_1$ and $Q_2$ here represent frame-level quantization choices, it does not imply that all blocks within the frame have the same quantization scale. Thus, $Q_1$ could consist of a vector choice of different quantization scales for each block of frame 1. In addition, note that for arbitrary choices of $w_1$ and $w_2$, we have a weighted mean-squared-error (MSE) criterion, which reduces to the conventional (unweighted) MSE measure when $w_1 = w_2 = 1$.

### C. Examples

We now provide examples of problems of interest that follow as special cases of (3) and (4); see Fig. 2.

*Example 1—Independent Coding:* The independent case seen in Section II-A is a special case of (3) and (4), where frame 2 does not depend on frame 1, i.e., $R_2(Q_1, Q_2) = R_2(Q_2)$ and $D_2(Q_1, Q_2) = D_2(Q_2)$. The independent coding case arises for intraframe coding as well as pyramidal coding without quantization feedback [5] (see Section IV-B-1).

*Example 2—Spatially Dependent Pyramid Coding with Quantization Feedback:* When $w_1 = 0, w_2 = 1$, as will be shown in Section IV, we have the case of a two-layer closed loop (quantization feedback) pyramid, where the bottom, or full-resolution, layer (layer 2) depends on the quantization choice of the top, or coarse-resolution, layer (layer 1). Note that (4) refers to a total bit rate constraint, and we solve the full-resolution quality optimization problem only. In addition, for a multiresolution coding environment, it may be necessary to throw in additional constraints on the top layer bit rate

$$R_1(Q_1) \leq R_1' \tag{5}$$

or quality

$$D_1(Q_1) \leq D_1'. \tag{6}$$

*Example 3—Temporally Dependent Coding:* This is the most general case where (3) and (4) apply without any restrictions. DPCM and MPEG come under this class.

### D. Solution Based on Lagrange Multipliers

The problem of (3) and (4) can be solved by introducing the Lagrangian cost $J$ corresponding to the Lagrange multiplier $\lambda \geq 0$ as in [4] as follows:

$$J_1(Q_1) = w_1 D_1(Q_1) + \lambda R_1(Q_1), \tag{7}$$

$$J_2(Q_1, Q_2) = w_2 D_2(Q_1, Q_2) + \lambda R_2(Q_1, Q_2) \tag{8}$$

and considering the following unconstrained minimization problem:

$$\min_{Q_1,Q_2} [J_1(Q_1) + J_2(Q_1, Q_2)]. \tag{9}$$

Then, by a direct extension of Theorem 1 of Shoham and Gersho in [4], the following result follows:

*Fact 1:* If $(Q_1^*, Q_2^*)$ solves the unconstrained problem of (9), then it also solves the constrained problem of (3) and (4) for the particular case of $R_{budget} = [R_1(Q_1^*) + R_2(Q_1^*, Q_2^*)]$.

*Proof:* See Appendix A.

The above result implies that if we solve the unconstrained problem of (9) for a fixed value of $\lambda$, and if the total bit rate happens to be $R_{budget}$, then we have also optimally solved the constrained optimization problem of (3) and (4). Further, as $\lambda$ is swept from 0 to $\infty$, one traces out the convex hull of the composite R-D curve of the dependent allocation problem. The monotonic relationship between $\lambda$ and the expended bit rate [4] makes it easy to search for the "correct" value of $\lambda$, say $\lambda^*$, for a desired $R_{budget}$.

Note how for the independent case ($J_2(Q_1, Q_2) = J_2(Q_2)$), where there is a single $R_2 - D_2$ curve in Fig. 1, (9) becomes the familiar result of [4]. Here, each frame is minimized independently, as was shown in Section II-A.

For the general dependent case, the two-frame problem becomes the search for $Q_1^*, Q_2^*$ that solve

$$J_1(Q_1^*) + J_2(Q_1^*, Q_2^*) = \min_{Q_1,Q_2} [J_1(Q_1) + J_2(Q_1, Q_2)] \tag{10}$$

$$= \min_{Q_1} [J_1(Q_1) + J_2(Q_1, Q_2^*(Q_1))] \tag{11}$$

where $J_2(Q_1, Q_2^*(Q_1)) = \min_{Q_2}[D_2(Q_1, Q_2) + \lambda R_2(Q_1, Q_2)]$ is the minimum Lagrangian cost (for quality condition $\lambda$) associated with the dependent frame when the independent frame is quantized with $Q_1$; see Fig. 1. Thus, for the desired operating quality $\lambda$, we find the optimal solution by finding, for all choices of $Q_1$ for the independent layer, the optimal $(Q_2^*(Q_1))$, which "lives" at absolute slope $\lambda$ on the (dependent) $R_2$-$D_2$ curve associated with $Q_1$.

By a simple extension of this result, it follows that the optimal solution to our general N-frame dependency problem consists in introducing $J_i(Q_1, Q_2, \ldots, Q_i) = w_i D_i(Q_1, Q_2, \ldots, Q_i) + \lambda R_i(Q_1, Q_2, \ldots, Q_i)$ for $i = 1, 2, \ldots, N$ and solving the following unconstrained problem for the "correct" value of $\lambda$ which meets the given $R_{budget}$:

$$\min_{Q_1,Q_2,\ldots,Q_N} [J_1(Q_1) + J_2(Q_1, Q_2) + \ldots J_N(Q_1, Q_2, \ldots, Q_N)]. \tag{12}$$

### E. Complexity

The optimal solution, as shown by (12) is obviously exponentially complex in the dependency-tree depth $N$. Moreover, it has to be pointed out that the computational complexity is dominated by the *data generation* phase, i.e., finding all the $(R_i(Q_1, Q_2, \ldots, Q_i), D_i(Q_1, Q_2, \ldots, Q_i))$ points for the problem is much more complex than finding the optimal solution, given all the possible R-D operating points. In order to ease this computational burden, we are therefore interested in methods which will avoid the need to grow all the R-D data, while retaining optimality. Note that while our methods rely on generating "real" R-D data, a model-based approach may be used within our framework to ease this burden. We now examine an important property which enables us to do this, and which will be used in Section III to formulate pruning conditions to eliminate suboptimal choices in the MPEG allocation problem.

### F. Monotonicity

The key to obtaining a fast solution to the complex dependent allocation problem of (3) and (4) is the monotonicity property of the R-D curves of the dependent components (frames). Consider the example of two frames, with the operational R-D curve of the second frame depending on that of the first, as in Fig. 1. Assume that the quantizer grades are ordered monotonically from finest to coarsest. Let us use $i$ and $j$ to denote the quantization choices for the independent and dependent frames, respectively.

*Definition 1:* A dependent coding system has the monotonicity property if, for any $\lambda \geq 0$:

$$J_2(i,j) \leq J_2(i',j), \text{for i} \leq \text{i}'. \tag{13}$$

For example, for $\lambda = 0$, this means that

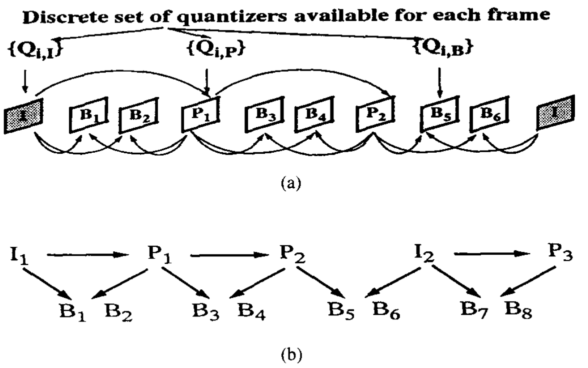
$$D_2(i,j) \leq D_2(i',j), \text{for i} \leq \text{i}'. \tag{14}$$

Stated in words, the monotonicity condition simply implies that a "better" (i.e., finer quantized) predictor will lead to more efficient coding, in the rate-distortion sense, of the residue (whose energy decreases as the predictor quality gets better), that is, the dependent frame's family of R-D curves will be monotonic in the fineness of the quantizer choice associated with the parent frame from which they are derived. As can be seen, the finer the quantization for frame 1 (Fig. 1(a)), the closer to the origin of the $R_2 - D_2$ graph will be the corresponding curve for the dependent frame 2 (Fig. 1(b)). Experimental results involving MPEG *verify this monotonicity property* for all the cases that we studied. Thus, monotonicity appears to be a realistic property, which has favorable theoretical implications as well, and it can be used to formulate fast pruning conditions for the MPEG allocation problem in Section III. In the rest of this work, we will address faster ways of attaining and approximating the optimal solution in the temporal and spatial dependent frameworks.

## III. Temporal Dependency: The MPEG Case

We now address the general temporal dependency quantization problem of which MPEG [7] is an example. The MPEG coding format, a CCITT video compression standard, shown in Fig. 3, segments the video sequence into identical groups-of-pictures (GOP's). Each GOP consists of three types of frames using the intraframe ($I$), prediction ($P$), and bidirectionally interpolated ($B$) modes of operation. The $I$ frames are coded independently, the $P$ frames are predicted from the previous $I$ or $P$, and the $B$ frames are interpolated from the previous and next $I$ and/or $P$ frames. For the two-layer dependency illustrated in Fig. 1, the problem we are trying to solve (for an MSE criterion) is the problem of (3) and (4) with $w_1 = w_2 = 1$:

$$\min_{Q_1,Q_2} [D_1(Q_1) + D_2(Q_1,Q_2)]$$
$$\text{s.t.} R_1(Q_1) + R_2(Q_1,Q_2) \leq R_{budget}.$$



(a)

(b)

Fig. 3. Typical MPEG coding framework: (a) MPEG frames: The $I$ frames are indepedently coded, the $P$ frames are predicted from previous $I$ or $P$ frames, and the $B$ frames are interpolated from adjacent $I$ and/or $P$ frame pairs; (b) temporal dependency in the MPEG framework. Note that the $B$ frames are leaves in the dependency tree.

The solution to this problem was shown in Section II-B as being exponentially complex in the dependency tree depth. Here, we will show how to reduce this complexity for the MPEG coding case (see Fig. 3). Before we tackle the general MPEG problem (with $I$,$P$, and $B$ frames), we begin with a simpler special case of MPEG that is easier to analyze and which provides the intuition for the more complex general problem.

### A. A Particular Case of MPEG: I-B-I

We consider a special case of MPEG having only $I$ and $B$ frames (see Fig. 4), i.e., the predicted $P$ frames of the more general MPEG format are omitted. The dependency tree is shown in the more compact form of a trellis. The "states" of the trellis represent the quantization choices for the *independently coded* $I$ frames (ordered from top to bottom in the direction of finest to coarsest), while the "branches" denote the quantizer choices associated with the two $B$ frames.

The trellis is populated with Lagrangian costs (for a fixed $\lambda$) associated with the quantizers for each frame. Let us focus on the $I_1 - B_1 - B_2 - I_2$ stage of the trellis. The state nodes are populated with the costs of the respective $I$ frame quantizers $J(Q) = (D(Q) + \lambda R(Q))$. Each $(i,j)$ branch connecting quantizer state $i$ of $I_1$ to quantizer state $j$ of $I_2$ is populated with the sum of the minimum Lagrangian costs of the $B_1$ and $B_2$ frames, i.e., with $J(B_1) + J(B_2)$, where

$$J(B_l) = \min_{Q_{B_l}}[D(Q_{B_l}) + \lambda R(Q_{B_l})] \text{ for } l = 1,2 \tag{15}$$

where the R-D curves for $B_1, B_2$, are generated from the $i,j$ quantizer choices for $I_1, I_2$, respectively. From (12), it is clear that the optimal path is that which has the minimum total cost across all trellis paths. Since the independent $I$ frames "decouple" the $B$ frame pairs from one another, it

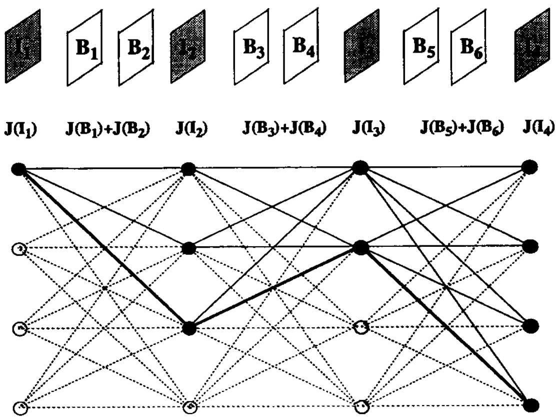


Fig. 4. *I-B-I* special case of MPEG, Finding an R-D convex hull point corresponding to a $\lambda$ is equivalent to finding the smallest cost path through the trellis. Each trellis node corresponds to a quantizer choice for the $I$ frames, monotonically ordered from finest to coarsest, and is populated with the associated Lagrangian cost ($J(I) = D(Q) + \lambda R(Q)$). The branches correspond to the $B$ frame pairs and are populated with their minimum Lagrangian costs ($J(B) = \min[D(q) + \lambda R(q)]$) for the particular $I$ frame quantizer choices given by each branch's end nodes. For quality slope $\lambda$, the optimal total cost path is obtained with the Viterbi algorithm. The "dark line" path joins the smallest-cost $I$ frame nodes. Monotonicity implies that all dashed line paths can be pruned out.

is obvious that the popular Viterbi algorithm (VA) [12] will provide the minimum cost path through the trellis, i.e. we need to keep a single path (the minimum cost one) arriving at each node. More formally, we have the following algorithm for the $I_1 - B_1 - B_2 - I_2$ stage of the trellis.

*Algorithm 1:*

**Step 1**: Generate $J(I_1)$ and $J(I_2)$ for $I$-frames $I_1$ and $I_2$ for all quantizers in $Q_{I_1}, Q_{I_2}$ respectively, i.e., populate all the **nodes** of the trellis with the Lagrangian costs.

**Step 2**: For every pair of nodes $(i,j)$, where $i$ is the quantizer choice for $I_1$ and $j$ for $I_2$, assign branch cost $J(B_1) + J(B_2)$, where $J(B_l)$ for $l = 1, 2$ are obtained from (15), i.e., populate all the **branches** of the trellis with the minimum Lagrangian costs.

**Step 3 (VA pruning rule)**: At every node of $I_2$, keep only that branch that minimizes $J(I_1) + J(B_1) + J(B_2)$.

As noted in Section II-E, the computational complexity is dominated by the *data generation* phase, i.e., in the trellis population phase.

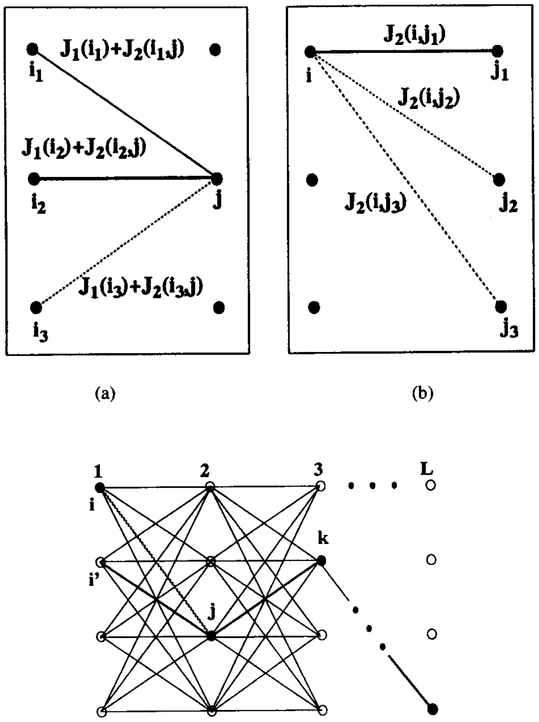
### *B. Pruning Conditions Implied by Monotonicity*

The monotonicity condition stated earlier in Section II-F will now be used to formulate pruning conditions to eliminate suboptimal operating points in the temporal dependency coding problem. The first lemma is associated with Fig. 5(a). As a reminder, the quantizer states are ordered monotonically from finest to coarsest.

*Lemma 1:* If

$$J_1(i) + J_2(i,j) < J_1(i') + J_2(i',j)\ \mathit{for}\ \mathbf{any}\ i < i' \qquad (16)$$

then the $(i',j)$ branch cannot be part of the optimal path and can be pruned out.



(a) (b)

(c)

Fig. 5. Pruning conditions obtained from monotonicity: (a) $J_1(i_2) + J_2(i_2,j)$ is the minimum Lagrangian cost of all branches terminating in node $j$. Therefore (see Lemma 1), the $(i_3,j)$ branch can be pruned; (b) $J_2(i,j_1))$ is the minimum Lagrangian cost of all branches originating from node $i$. Therefore (see Lemma 2), the $(i,j_2)$ and the $(i,j_3)$ branches can be pruned; (c) diagram used for the proof of Lemma 1.

*Proof:* We prove the lemma by contradiction. Assume that $(i',j)$ for any $i < i'$ **is** part of the optimal path (see Fig. 5(c)). Let the optimal quantizer sequence path be $(i',j,k,\ldots,l)$. However, by monotonicity, we have

$$J_3(i,j,k) \le J_3(i',j,k) \qquad (17)$$

$$\cdots$$

$$J_L(i,j,k,\ldots,l) \le J_L(i',j,k,\ldots,l). \qquad (18)$$

Summing up (16), (17), ..., (18), we get the contradiction that the total Lagrangian cost of the path $(i,j,k,\ldots,l)$ is smaller than that of the optimal path $(i',j,k,\ldots,l)$. □

The above lemma is associated with pruning branches that merge into a common destination state. A dual result holds for the pruning of branches that originate from a common source state (see Fig. 5(b)) leading to the following companion Lemma 2, whose proof is omitted as it is similar to that of Lemma 1.

*Lemma 2:* If $J_2(i,j) < J_2(i,j')$ for any $j < j'$,then the $(i,j')$ branch cannot be part of the optimal path and can be pruned out.

Note that a consequence of the above lemma is that if $J_1(i) < J_1(i')$ for $i < i'$, then the state node $i'$ (and all branches from it) can be pruned out. The two pruning conditions of Lemmas 1 and 2 can be used to lower the

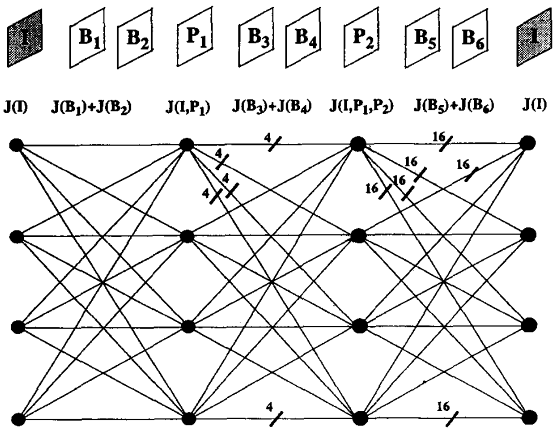


Fig. 6. General MPEG "trellis" diagram extension of Fig. 3. Here, the inclusion of the $P$ frames prevents the decoupling of the $B$ frame pairs, and the entire tree has to be grown. Note that each stage of the trellis is represented by "vector" branches whose dimension grows exponentially with the dependency tree depth.

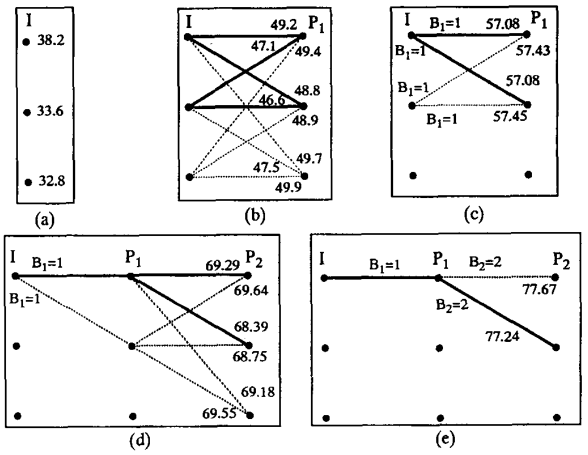


Fig. 7. Tree pruning using the monotonicity property (Lemmas 1 and 2). The numbers are the cumulative Lagrangian costs for a typical example for $\lambda = 10$. Branches pruned at each stage are shown with dashed lines. In this example, the number of R-D points generated is cut down from 363 (exhaustive) to only 36 with no loss of optimality (40.76 dB at 1 bpp) using Algorithm 2.

complexity of the VA-based search. In the special case of MPEG of Section III-A (refer to Fig. 4), Lemmas 1 and 2 eliminate the need to consider the full trellis on which to run the VA, making it unnecessary to consider any paths lying below the (dark line) path connecting the minimum cost state nodes of the $I$ frames, i.e., in Step 1 of Algorithm 1, all $I$-frame nodes lying below the minimum cost node can be eliminated. This is because any path with excursions below the path connecting the minimum cost state nodes (corresponding to the $I$ frames only) can be replaced by one which lies *above* this boundary, by monotonicity.

### C. General MPEG Bit Allocation

Having established the intuition behind dependent allocation and the power of monotonicity, we now evolve to the more complex (general) MPEG format of Fig. 6. The presence of the $P$ frames extends the dependency tree depth, and the decoupling between successive stages of the trellis is lost. We can thus no longer resort to the Viterbi algorithm, but must instead retain the entire tree, which grows exponentially with the number of dependent levels. The good news, however, is that the monotonicity conditions still apply, and the pruning conditions of Lemmas 1 and 2 can aid in reducing complexity dramatically. As an example, see Fig. 7 where we consider an $I$-$B$-$P$-$B$-$P$ sequence of MPEG frames (note that for simplicity, we use only one $B$-frame between $I$-$P$ pairs) and a choice of three quantizer grades for each frame. More formally, the algorithm used is the following (refer to Figs. 6 and 7).

*Algorithm 2:*

**Step 1**: Generate $J(I)$ for all quantizers $q \in Q_I$; see Fig. 7(a).

**Step 2**: *(Monotonicity)* Prune out all $I$-nodes lying below minimum cost node $q^* \in Q_I$ in Step 1.

**Step 3**: Grow $J(I, P_1)$ for all combinations of $q \in Q_{P_1}$ and all remaining $q \in Q_I$ after Step 2. See Fig. 7(b).

**Step 4**: *(Monotonicity)* Use pruning conditions of Lemmas 1 and 2 to eliminate suboptimal $I - P_1$ combinations; see Fig. 7(b).

**Step 5**: For every surviving $I - P_1$ combination, find the $B_1, B_2$ quantizer pair that minimizes $J(B_1) + J(B_2)$, i.e., populate the branch costs of the trellis of Fig. 6. See Fig. 7(c).

**Step 6**: *(Monotonicity)* Use pruning conditions of Lemmas 1 and 2 to eliminate suboptimal $I - B_1 - B_2 - P_1$ combinations; see Fig. 7(c).

**Step 7**: For all remaining paths, repeat Steps 3 to 6 for the $(P_1 - B_3 - B_4 - P_2)$ and the $(P_2 - B_5 - B_6 - I)$ sets.

The smallest cost path after running Algorithm 2 is the optimal solution corresponding to the chosen $\lambda$ for the group-of-pictures considered. While the exhaustive search would have us grow as many as 363 Lagrangian costs, in our example, only 36 costs need to be grown, an order of magnitude reduction in complexity with no loss of optimality if the monotonicity conditions apply (in our example, application of Algorithm 2 for $\lambda = 10$ provides an optimal R-D operating point — 40.76 dB at 1 bpp — as was verified through exhaustive search). The complexity reduction due to monotonicity is dependent on the desired quality slope $\lambda$, with higher quality targets achieving better reduction. In the limit, as $\lambda$ goes to 0, the minimum cost path is always the one corresponding to the finest quantizers and thus only a single "highest quality" path has to be grown. Conversely, if $\lambda$ goes to $\infty$ the monotonicity property provides no gain.

*1) Suboptimal Heuristics:* As pointed out, the amount to which the monotonicity property can be exploited is $\lambda$ dependent and may not suffice for some applications. To this end, it is advisable to come up with fast heuristics, which, used in combination with monotonicity, can approach the optimal performance at a fraction of the complexity. In try-

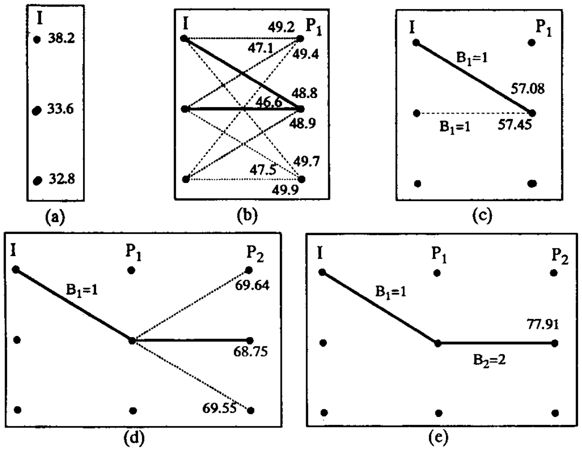


Fig. 8. Tree pruning using monotonicity as well as a "greedy" heuristic for the same conditions as those of Fig.˜6. The number of R-D points generated is now 24, at a slight loss of optimality (total Lagrangian cost is 77.91 versus optimal cost of 77.24–40.76 dB at 1 bpp) using Algorithm 2 with the heuristics of Section III-C-1.

ing to formulate a fast MPEG heuristic, it is necessary to consider some important points: i) the "anchor" $I$-frame is the most important of the group of pictures and must not be compromised, ii) most signal sequences enjoy a finite memory property, where the influence of a parent frame diminishes with the level of its dependency.

Thus, it may pay to choose (only) the lowest cost nodes for all frames except the $I$ frame, for which we retain all nodes remaining after applying monotonicity-based pruning. Thus, a single path is grown from each of these admissible $I$-frame nodes, whereas in the general case, a whole tree evolves from each such node. Based on this, we propose the following heuristic: i) retain all paths that *originate from* each of the $I$-frame quantization states remaining after the monotonicity-based pruning, i.e. it is not prudent to be greedy for the $I$-frame, as a greedy error affects all dependent frames derived from it; ii) use a "greedy" pruning condition (in combination with the monotonicity property) to keep only the lowest cost branch thus far at all other stages in the trellis, that is, we follow Algorithm 2, except that we add an extra pruning condition in Steps 4 and 6, where we retain only a single (minimum cost) path corresponding to every surviving $I$-frame node. This heuristic, as shown in Fig. 8, leads to near optimal performance at a fraction of the computational cost. For the example of Figs. 7 and 8, the optimal solution for a particular $\lambda$ gives 40.76 dB at 1 bpp, while the heuristic achieves 40.45 dB at 0.97 bpp, certainly very close to optimality. Unlike Algorithm 2, which relies on monotonicity pruning conditions only and which works best when $\lambda$ is low (high quality), the fast heuristic retains low complexity even at high values of $\lambda$. Note how choosing the greedy solution for the $I$-frame (bottom-most node) would have given a much worse performance.

*2) Discussion:* We have shown a method to find the optimal bit allocation strategy for an MPEG coding framework, assuming arbitrary quantizer sets for each MPEG frame. Although our scheme can be computationally complex, it can serve as an optimal benchmark to evaluate more practical allocation strategies. In addition, model-based approaches (e.g., measuring one R-D point and using a model-based extrapolation to find other points) can be combined with our techniques to ease computational burden. Note that since MPEG coders are typically buffered with a buffer size of the order of a group-of-pictures, formulating a bit allocation strategy for more than a GOP is somewhat irrelevant, thus increasing the practicality of our approach.

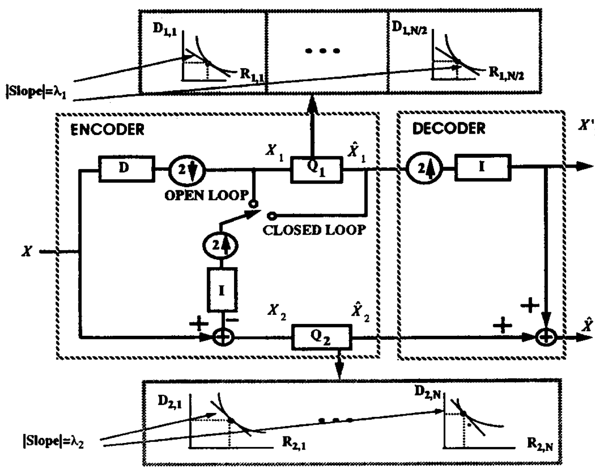


Fig. 9. Two-layer Laplacian pyramid. Open-loop mode bypasses the coarse level quantizer $Q_1$, whereas closed-loop mode includes it.$X$ is the source, $X_1$ the half-resolution signal, $\hat{X}_1$ its quantized version,$X_1^{prime}$ the full-resolution quantized coarse approximation, $X_2$ the residue signal, $\hat{X}_2$ its quantized version, and $\hat{X}$ the full-resolution quantized output.

## IV. Multiresolution Coding Using A Pyramid

We have dealt with the temporal dependency case in Section III. We now proceed to the pyramid-based multiresolution coding framework, where we study the spatial and spatio-temporal pyramids. Refer to Fig. 9 for a two-layer Laplacian pyramid coder [5] which we use, without loss of generality, as a tool for multiresolution analysis. Note that $D_1$ and $D_2$ now refer to the top layer and bottom layer distortions respectively. The top layer will be referred to as the **coarse or subresolution layer**, while the bottom will be referred to as the **difference or residue layer**. The coarse resolution added to the detail resolution gives the full-resolution signal.

### A. Motivation

We will now pose the MR problem description in terms of the general formulation of Section II-B (seen briefly in Example 2). Here, we elaborate on the need for the additional subresolution constraints (5) or (6) in solving the general problem of (3) and (4). As shown in Fig. 9, in this section, we analyze both closed- and open-loop modes of operation. As will be shown, the closed-loop system outperforms the open-loop system and we will therefore confine ourselves to the former in specifying several useful problem definitions. The following scenarios are of interest for the closed-loop system (refer to Fig. 9):

**a)** minimize the full-resolution distortion $D_2(Q_1, Q_2)$ under a total bit rate constraint of $(R_1(Q_1) + R_2(Q_1, Q_2)) \leq R_{budget}$ bits to code both resolutions, i.e., the special case of (3) and (4) with $w_1 = 0, w_2 = 1$;

**b)** minimize the full-resolution distortion $D_2(Q_1, Q_2)$ under the constraints of both total bit rate $(R_1(Q_1) + R_2(Q_1, Q_2)) \leq R_{budget}$ and a maximum tolerable coarse resolution distortion $D_1(Q_1) \leq D_1'$, i.e., case **(a)** with the added constraint of (6);

**c)** minimize a distortion metric which is a weighted average of the full-resolution and coarse-resolution distortions $w_1 D_1(Q_1) + w_2 D_2(Q_1, Q_2)$ under the constraints of total bit rate for both resolutions as well as a maximum bit rate for the coarse resolution $R_1(Q_1) \leq R_1'$, i.e., the general case of (3) and (4) with the added constraint of (5).

The problem of **a)** disregards any compatibility requirements, and aims at optimizing the full-resolution quality only. Thus, for compatible coding problems, the coarse-resolution quality that comes for free as a "by-product" of minimizing the full-resolution distortion may not be acceptable. For most applications, the optimal allocation of **a)** usually results in an unacceptable subresolution quality, necessitating a formulation like **b)** or **c)**. The problems of **b)** and **c)** involve compatible subchannel allocations and are duals of each other. In this paper, we will focus on **b)**, with results applying to **c)** by duality.

### *B. Spatial Pyramid*

We first analyze the spatial pyramid bit allocation to motivate the spatio-temporal video pyramid bit allocation problem to be tackled later. For the image pyramid, we consider signal units which are blocks (e.g., 8x8 blocks of JPEG) with additive rate and distortion functions. Note that we do a 2-D analysis (subsampling by four per layer) and the block sizes are the same at each layer.

$$D_1(Q_1) = \sum_{j=1}^{N/4} d_{1,j}, \qquad R_1(Q_1) = \sum_{j=1}^{N/4} r_{1,j}, \tag{19}$$

$$D_2(Q_1, Q_2) = \sum_{j=1}^{N} d_{2,j}, \qquad R_2(Q_1, Q_2) = \sum_{j=1}^{N} r_{2,j}. \tag{20}$$

The bit allocation at the block level for the image pyramid will motivate the bit allocation at the frame level of Section IV-C, where we will use frames instead of blocks as signal units. We will use $q, d, r$ and $Q, D, R$ respectively for block level and frame level parameters.

*1) Open-Loop Pyramid:* Fig. 9 shows a two-layer pyramid. Here, we assume open loop operation, i.e. the feedback prediction loop bypasses the subsampled layer's quantizer, as introduced in [5].

*a) Bit Allocation for Optimizing Full-Resolution Quality:* We want to determine the optimal bit allocation (for a given total bit budget) that maximizes the full-resolution quality. This problem for the special case of an ideal half-band lowpass filter, and negative exponential scalar quantizers has been studied in [13], under the assumption that the total squared-error distortion is the sum of the individual squared-error distortions of the layers of the pyramid (assuming normalized filters, i.e. filters which preserve energy). Under the same conditions, we can generalize the result to arbitrary quantizer sets; see Fig. 9.

Assuming that each independently coded block $i$ has its own set of admissible quantizer choices $\{q_i\}$, under the additivity of distortion assumption, the open loop problem becomes the minimization of $\mathcal{D}(X, \hat{X}) = \mathcal{D}(X_1, \hat{X}_1) + \mathcal{D}(X_2, \hat{X}_2)$, i.e. the solution to the independent allocation problem of (1) and (2) seen in Section II-A, The solution is that at optimality each block $i$ of the coarse (subsampled) layer should be operating at the same *slope* $\lambda_1$ on its operational R-D curve as each block $j$ of the residue layer i.e., $\lambda_1 = \lambda_2$ in Fig. 9. This is a simple extension to generalized quantizer sets of the 1-D signal equivalent result of [13] (for normalized filters) that at optimality the coarse layer should have the same *distortion* as the residue layer.

*b) Compatible Subchannel Allocation:* If we now add subresolution constraints, our problem becomes:

$$\min D_1(Q_1) + D_2(Q_2) \text{ s.t. } R_1(Q_1) + R_2(Q_2) \leq R_{budget} \tag{21}$$

$$\text{and } D_1(Q_1) \leq D_1'. \tag{22}$$

Obviously, if finding the full-resolution optimal solution of (21) meets the subresolution constraint $D_1(Q_1) \leq D_1'$ of (22), then we are done. However, if it does not, then it can be shown that at optimality, the coarse resolution allocation should *just* meet the subresolution constraint, i.e., we have the following Lemma.

*Lemma 3:* The optimal solution to (21) and (22) is to choose $R_1^*$ so that $D_1^* = D_1'$ and then allocate $R_2^* = R_{budget} - R_1^*$ for the bottom layer.

*Proof:* See Appendix B.

This is an important result, for in the limit of fine quantization for the coarse layer (i.e., as $Q_1$ gets finer), the open-loop and closed-loop solutions converge. Thus, this (optimal) allocation strategy for the open-loop case would also be an effective one for the closed-loop case, at least in the limit of fine quantization of the coarse layer.

*2) Closed-Loop Pyramid:* We now analyze the closed loop case (see Fig. 9), which includes the quantizer in the prediction feedback loop of the pyramidal coder, as in [6]. The closed loop allocation is different because a) the full-resolution distortion $\mathcal{D}(X, \hat{X})$ is equal to the residue layer distortion $\mathcal{D}(X_2, \hat{X}_2)$, due to the feedback "absorbing" the quantization distortion of layer 1, and b) the unquantized residue layer signal depends on the choice of the coarse level quantizer $Q_1$, i.e., $X_2 = f(Q_1)$.

*a) Bit Allocation for Optimizing Full-Resolution Quality:* We now tackle case **a)** of Section IV-A. The problem, as a reminder, is to solve

$$\min D_2(Q_1, Q_2) \; s.t. \; R_1(Q_1)+R_2(Q_1, Q_2) \leq R_{budget}. \quad (23)$$

See Fig. 9.

This problem is a special case of the generalized problem of (3) and (4) with $w_1 = 0$ and $w_2 = 1$ and was solved in Section II-A. As was seen, the solution consists of using a Lagrangian formulation. For a fixed $\lambda$, we find the minimum total Lagrangian cost $J(Q_1, Q_2)$ by finding the $(Q_2^*(Q_1))$, which "lives" at absolute slope $\lambda$ on the (dependent) $R_2$-$D_2$ curve associated with each quantization choice $Q_1$ of the independent layer.

The analysis above solves the problem at a "layer" or "frame" level, so that $Q_1$ and $Q_2$ are quantization choices for the entire layer or frame, e.g., using a JPEG [14] coding framework. If we specify the problem at the block level, i.e., if we solve

$$\min \sum_{i=1}^{N} d_{2,i} \; s.t. \; \sum_{i=1}^{N/4} r_{1,i} + \sum_{i=1}^{N} r_{2,i} \leq R_{budget} \quad (24)$$

then the number of choices for the optimal solution make it impractical to be solved by brute force, but it can be shown that at optimality, all *blocks* in the dependent (residual) layer must operate at a constant slope, though no such condition is necessary for the independent (coarse) layer blocks. The difficulty with optimality at the block level comes from the inter-block dependencies due to the filtering operation in the interpolation process. If however we limit our dependency to parent/children sets (with a subresolution "parent" block being coupled with its directly interpolated four children blocks—see Fig. 10), then a fast way to approach the optimal solution (using Lagrangian methods) is feasible. This is accomplished (see Appendix C for details) by treating the parent/children sets as independent entities over which to apply bit allocation techniques, using the "constant slopes at optimality" principle. As the interpolation filter kernel gets shorter with respect to the block size, this approximation gets better, since the second-order "leakage" effect of inter-block dependencies gets smaller. In the limit, for the trivial two-tap averaging filter, our analysis becomes exact.

*b) Compatible Subchannel Allocation:* The noncompatible optimization of the previous section, as borne out by experimental evidence, may not provide a "usable" subchannel quality, if there should be such a constraint. Thus, optimizing for only the full-resolution quality as in (23) is not only computationally complex, but also not useful! It may be more profitable to sacrifice the full-resolution quality slightly (typically by < 0.3 dB) to gain in the coarse-resolution quality (typically by about 3 dB), for a given total budget. Moreover, this can be achieved very efficiently using the following fast algorithm which imposes the constant-slopes operating point condition for the blocks within each layer of the pyramid. The idea is to operate at constant slope $\lambda_1$ for each block of the coarse layer such that the distortion quality constraint is **met or exceeded**, and to allocate the remaining bits among the blocks in the residue layer again in a constant slope fashion. This strategy guarantees that the coarse layer allocation is optimal, i.e., one cannot achieve the distortion performance for the coarse layer with fewer bits. In addition, it separates the difficult optimization problem into separate fast minimizations over each pyramid layer.

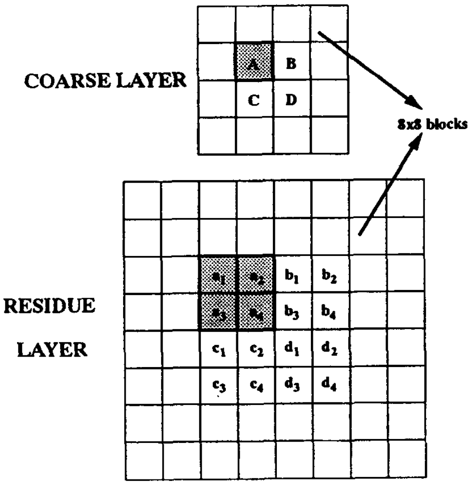


Fig. 10. Parent-child dependency for the two-layer 2-D pyramid. Assuming that the same block size (8x8) is used for both the coarse and residue layers, the filtering effect for short kernel filters will cause the dependency scope of each parent block (e.g. $A$) of the coarse layer to be essentially confined to its four children (e.g., $a_1, a_2, a_3, a_4$) of the residue layer.

The idea of the algorithm is to map out, in the most general case, all constant-slope operating points for each pyramid layer for a given total bit budget. Note that although the algorithm exhaustively searches for all possible points, in practice, this is not necessary. A formal summary of the algorithm follows:

*Algorithm 3:*

**Step 1**: Find the (constant) operating slope $\lambda_1$ for all blocks of the coarse layer which **just meets** (to within a convex hull approximation, as in [11]) using $R_1$ bits, the given coarse layer distortion requirement: $\mathcal{D}(X_1, \hat{X}_1) \leq D_1'$. If $R_1 > R_{budget}$, then stop. We have no more admissible coarse layer (convex hull) operating points. Else, go to Step 2.

**Step 2**: For the $\lambda_1$ of Step 1, generate $X_2$ and its associated $R_2$-$D_2$ curve. Operate each block of $X_2$ at constant slope $\lambda_2 \geq 0$, which meets the budget $R_2 \leq (R_{budget} - R_1)$. If no such $\lambda_2$ exists, stop. We have no more admissible residue layer (convex hull) operating points.

**Step 3**: Measure $\mathcal{D}(X, \hat{X}) = \mathcal{D}(X_2, \hat{X}_2)$. If it is smaller than the previous iteration's, it becomes the current candidate for the optimal solution: update $\lambda_1^* = \lambda_1$,$\lambda_2^* = \lambda_2$.

**Step 4**: Decrease $\lambda_1$ to the next higher-quality operating point and repeat Steps 1 through 3, until that slope $\lambda_{1,min}$ is reached for which either the given budget $R_{budget}$ is exhausted in the coarse layer in Step 1, or the coarse layer quantizer $Q_1$ is at its finest permissible value ($R_1(Q_1)$ is maximum). The optimal operating slopes are $\lambda_1^*$,$\lambda_2^*$.

The complexity of Algorithm 3 obviously increases exponentially with the number of pyramid layers. However, experimental results indicate that for typical images, filters, and quantizer choices corresponding to "usable" quality sub-

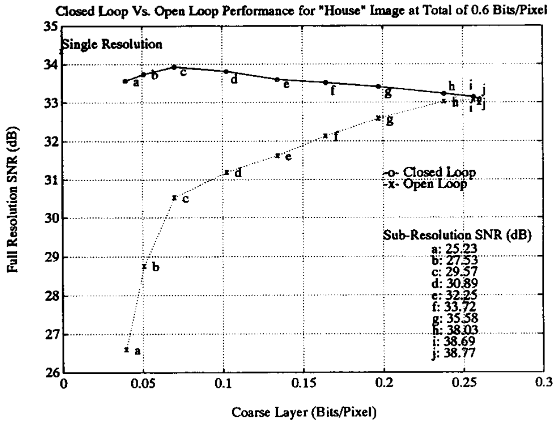


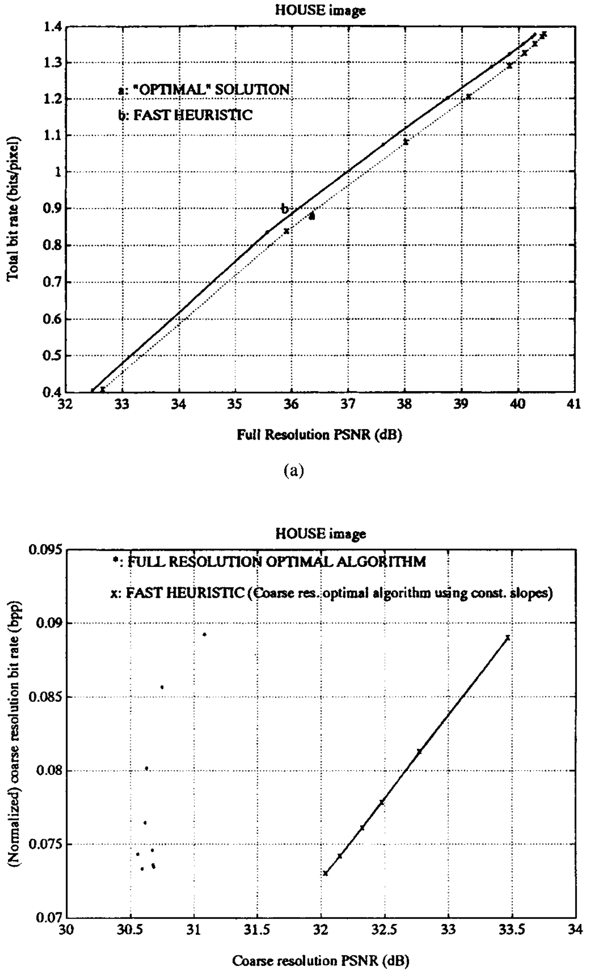
Fig. 11. Comparison in performance of the open- and closed-loop pyramidal coding schemes for the "House" image. The subresolution SNR's for the compatible subchannel are labeled from a to j. Note, for example, that at a subresolution bit rate of about 0.1 bpp corresponding to a coarse level SNR of 30.89 dB (point d), the closed- loop scheme outperforms the open-loop scheme by nearly 3 dB. The gain of single resolution over the closed loop is seen to be less than 1 dB. A JPEG-like coding environment is used with four admissible quantization scales (0.5,4.5,8,12) and (0.25,2.25,4,6) for the residue and coarse pyramid levels, respectively. The total bit budget is 0.6 bpp.

resolution signals, *a single iteration* is usually optimal, i.e., at optimality, the subresolution signal operates at the minimum bit allocation which *just meets* the desired coarse quality requirement. This was seen to be the optimal allocation for the open loop case in Section IV-B-1, and in the limit of fine coarse-layer quantization, as the open-loop and closed-loop schemes converge, this is seen to be an optimal operating point for the closed-loop scheme as well. In our experiments involving a five-tap Burt–Adelson filter and a JPEG coding framework, we found that "usable" subresolution images require fine coarse-layer quantization. Thus, in practice, a single iteration suffices, resulting in much-reduced complexity, leading to a *linear* increase in complexity with the number of pyramidal layers.

*3) Still Image Pyramid Coding Results:* We now describe the performance of several schemes in experiments using a modified JPEG coder for the pyramid layers, allowing each 8x8 block of each layer to have a quantizer scale choice from the discrete sets shown in Fig. 11. The results apply for the commonly-used "House" image.

*a) Closed Loop versus Open Loop Pyramid:* Fig. 11 shows a comparison in the performance of the open-loop and the closed-loop cases. The closed-loop curve was obtained using Algorithm 3. As can be seen, nearly 3 dB of subresolution quality gain can be obtained using the closed loop pyramid for the same full-resolution quality and the same total bit budget. This result highlights the benefits of operating closed-loop for most practical coding scenarios as in [6]. Note that in the limit of fine quantization of the coarse layer, the performances converge, as expected.

*b) Closed-Loop Coding Results—Optimal versus Fast Heuristic:* Fig. 12 shows a comparison of the results of the optimal allocation versus the constant-slope fast heuristic.



(a)

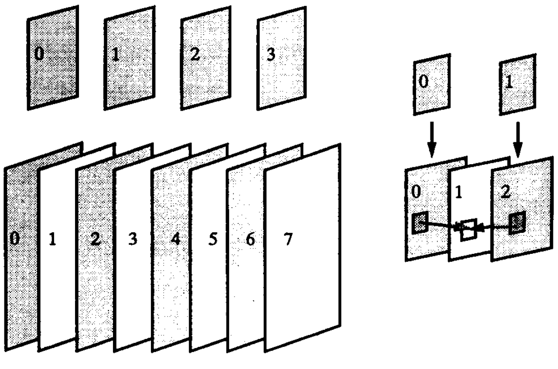
(b)

Fig. 12. Comparison of optimal (for full-resolution signal only) versus fast heuristic for a nine-tap interpolation filter. Note that for (a) the full resolution image, the optimal algorithm gains about 0.3 dB over the heuristic; for (b) the subsampled coarse resolution signal, the heuristic gains about 3 dB in SNR.

Both schemes use the same bit budget for each pyramid layer. As can be seen, the optimal allocation beats the heuristic by less than 0.3 dB in SNR for most bit rates of interest. Note, however, that the heuristic outperforms the optimal in the coarse resolution subsampled image (for which the heuristic is optimal) by about 2–3 dB in SNR. Note that this SNR improvement is in the subsampled image quality between the unquantized and quantized subsampled images (i.e., $X_1$ and $\hat{X}_1$ in Fig. 9). Similar improvement was seen in the upsampled coarse-resolution SNR's when the unquantized upsampled coarse signal was used as the reference.

### *C. Spatio-Temporal Video Pyramid*

Having laid the groundwork for the spatial pyramid allocation, we now address the spatio-temporal pyramid allocation of [6]. The MR video coder (see Fig. 13) uses a 3-D pyra-



(a) (b)

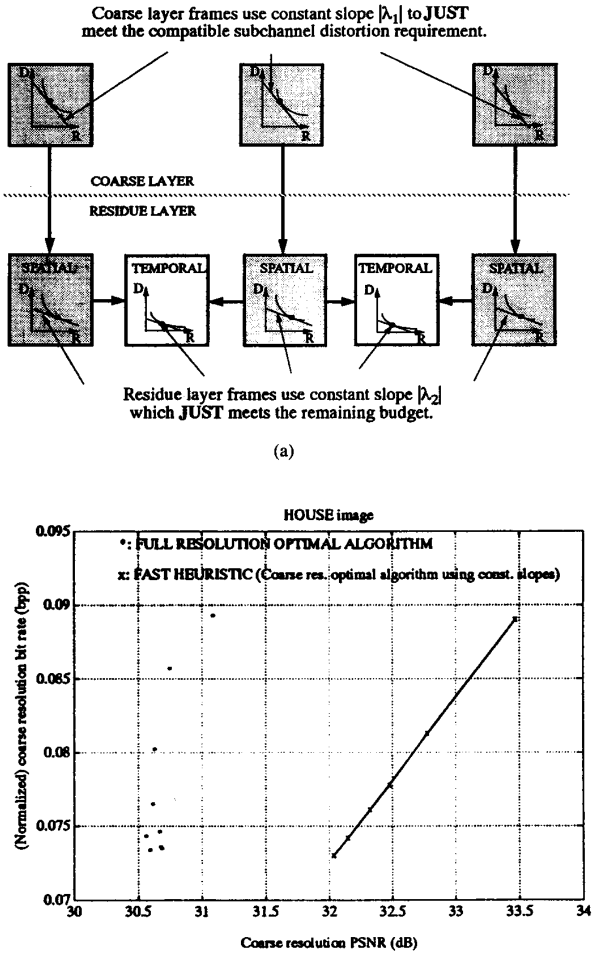
Fig. 13. (a) Spatio-temporal pyramid. Note that one half of the frames in the residue layer (shown as shaded) are spatially coded/interpolated; (b) one step of coarse-to-fine scale change in the reconstruction process. Note that frame 1 is temporally interpolated from frames 0 and 2.

midal decomposition, based on spatio-temporal interpolation, forming a hierarchy of video signals at increasing temporal and spatial resolutions. The structure is formed in a bottom-up manner, starting from the finest resolution, and obtaining a hierarchy of lower resolution versions. Spatially, images are subsampled after antialiasing filtering. Temporally, the reduction is achieved by simple frame skipping. For our analysis, we consider a two-layer pyramid. See Fig. 14(a) for the spatio-temporal dependency.

By invoking a hierarchical approach, one could directly extend the results of the spatial pyramid allocation described in Section III to the video pyramid allocation problem. Signal units change from blocks to entire frames of coarse and residue layers. The hierarchical approach consists in letting the set of admissible R-D operating points for each frame come from a constant-slope allocation of the blocks within each frame. This would lead to a set of admissible operating quality points which are optimal at the intraframe level and which span the entire range of bit rates of interest. The only difference in following this hierarchical extension to frames is that we now have to deal with temporally dependent frames in the residue layer of the video pyramid, where the (odd) temporal residue frames are dependent on the (even) spatial residue frames [6] (see Fig. 14(a)).

This is the direct extension to video coding of Algorithm 3 of the spatial case, i.e., operate all coarse frames at constant slope $\lambda_1^*$, and all residue frames, *temporal and spatial*, at constant slope $\lambda_2^*$. Note that Algorithm 3 is used with a single iteration, i.e., we operate the coarse layer frames at that slope $\lambda_1$ that *just meets* the desired subresolution rate or distortion constraint. Thus, Algorithm 3 separates the problem into a coarse layer allocation which is just an independent allocation problem, and a residue layer allocation. The latter is similar to the *I-B-I* temporal allocation problem of MPEG covered in Section III-A, where now the spatial residue frames play the role of the *I* frames and the temporal frames play the role of the *B* frames. Similar techniques to those described in Section III-A obviously apply here as well. Further, in this case, as we deal with residue images, the fast heuristic of operating at constant slopes for the spatial and temporal residue layers (i.e., choosing the minimal Lagrangian-cost nodes at every stage, which we depict as the "dark line" path of Fig. 4) was found to be always optimal in our experiments. The "greedy" constant slopes path (optimal if only spatial residue frames are considered) works well due to the big disparity in residual energy between the spatial and temporal frames. Fig. 15 shows experimental results for the ''mit'' sequence which confirm the remarkable performance of the constant slopes algorithm of Fig. 14(a).

*1) Video Pyramid Coding Results:* The allocation technique of Fig. 14 (b) was applied to the "mit" sequence with favorable results, as borne out objectively by the R-D curve of Fig. 15 (notice how no constant-slope point is suboptimal, i.e., all dark ''squares'' find themselves on



(a)

(b)

Fig. 14. (a) Dependency tree of the two-layer Laplacian spatio-temporal pyramid of [1] used as the MR video coder in this work and efficient allocation using constant slopes across the frames of each pyramid layer. Note that this is just the extension to video coding of Algorithm 1 using a single iteration; (b) example of allocation in the video pyramid.$\lambda_1, \lambda_2$, and $\lambda_3$ are quality factors for the coarse, spatial, and temporal interpolated frames, respectively.

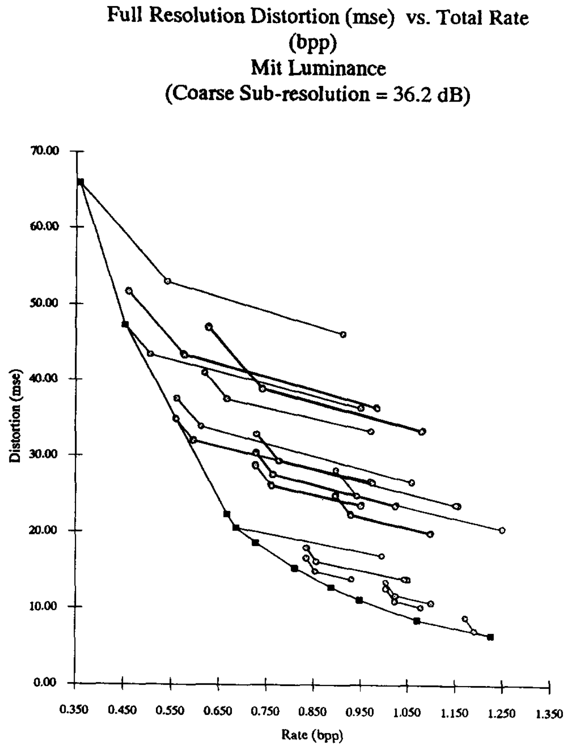


Fig. 15. R-D curve for residue layer of "mit" sequence. The subresolution quality is fixed at 28 dB. The dark dots show a constant-slope allocation among residual frames (spatial and temporal), whereas the other dots show the nonconstant slope points. Notice how no constant-slope point is suboptimal, i.e., all dark dots find themselves on the convex hull.

the convex hull). Various allocations and their performances are shown in Fig. 14(b). It can be seen that i) nonconstant slopes gives poor results, ii) the constant slope heuristic gives good coarse and full resolution quality, and iii) the full-resolution optimized allocation, which is computationally complex, gives marginally better full resolution quality than ii) but unacceptable coarse quality.

## APPENDIX A
## PROOF OF FACT 1

Note that in the proof, in the interest of notation brevity, we omit the explicit dependence of $D_1, R_1, J_1$ and $D_2, R_2, J_2$ on $Q_1$ and $Q_2$, i.e., we use $J_1 = J_1(Q_1), D_1 = D_1(Q_1), R_1 = R_1(Q_1)$ and $J_2 = J_2(Q_1, Q_2), D_2 = D_2(Q_1, Q_2), R_2 = R_2(Q_1, Q_2)$ and similarly for the optimal quantizers $Q_1^*$ and $Q_2^*$. Then, for all $Q_1, Q_2$, we have from the unconstrained minimization

$$J_1^* + J_2^* \le J_1 + J_2. \tag{25}$$

i.e.

$$D_1^* + \lambda R_1^* + D_2^* + \lambda R_2^* \le D_1 + \lambda R_1 + D_2 + \lambda R_2. \tag{26}$$

or

$$[D_1^* + D_2^*] - [D_1 + D_2] \le \lambda([R_1 + R_2] - [R_1^* + R_2 *]). \tag{27}$$

Since (27) holds for all admissible $\{Q_1, Q_2\}$, it certainly holds for the subset of $\{Q_1, Q_2\}$ for which $[R_1 + R_2] \le R_{budget}$, where $R_{budget} = [R_1^* + R_2^*]$. Therefore, from (27), since $\lambda \ge 0$, we have that

$$[D_1(Q_1^*) + D_2(Q_1^*, Q_2^*)] - [D_1(Q_1) + D_2(Q_1, Q_2)] \le 0, \tag{28}$$

i.e., over all $\{Q_1, Q_2\}$ that meet the rate budget, $(Q_1^*, Q_2^*)$gives the minimum distortion. □

## APPENDIX B
## PROOF OF LEMMA 3

Assume $\lambda^{ind}$ is the optimal slope in the solution to (21). Now, because $D_1^{ind} > D_1'$, the solution generated by $\lambda^{ind}$ is no longer feasible. This is equivalent to having the $R_1 - D_1$ characteristic truncated for values over $D_1'$, and solving the problem of (21) with this new R-D curve. Again, due to independence, at optimality both layers will have to be operated at a constant slope, say $\lambda'$. Because the allocation for the bottom layer $R_2$ has to decrease (otherwise we could not decrease the distortion $D_1$ of the top layer for fixed $R_{budget}$) we must have $\lambda' > \lambda^{ind}$ (as $\lambda$ increases, $R$ decreases and $D$ increases). In the original $R_1 - D_1$ curve, $\lambda'$ would result in a quality level such that $D_1(\lambda') > D_1^{ind}$ (increasing $\lambda$ increases the distortion) but since all points with $D_1 > D_1'$ have been removed we have that $D_1(\lambda') = D_1'$ (the highest available distortion). Therefore the optimal solution is indeed $D_1^* = D_1(\lambda') = D_1'$. □

## APPENDIX C
## BLOCK-LEVEL OPTIMAL FULL-RESOLUTION ALLOCATION FOR FOR TWO-LAYER CLOSED-LOOP PYRAMID (SECTION IV-B-2-i)

If we approximate the dependency scope of the filtering operation in the interpolation process of the spatial pyramid to direct parent/child dependencies, as shown in Fig. 10, we show how to get the optimal block-level allocation without exhaustive search, using a fast minimization algorithm where each parent/child "set" is treated independently.

Let there be $N/4$ coarse layer and $N$ residue layer blocks, all of the same size (see Fig. 10). Let the set of quantizer choices for the coarse layer 1, block $i$ be $\mathcal{Q}_i^p$ (for parent block), and the set for the residue layer 2, block $j$ be $\mathcal{Q}_j^c$ (for child block). Let $\mathcal{Q}^{(P)} = \mathcal{Q}_1^p X \mathcal{Q}_2^p X \ldots \mathcal{Q}_{N/4}^p$ be the set of all layer 1 block quantizer choices, and $\mathcal{Q}^{(C)} = \mathcal{Q}_1^c X \mathcal{Q}_2^c X \ldots \mathcal{Q}_N^c$ be the set of all layer-2 block quantizer choices. Then, the problem to solve is

$$\min_{Q_1 \in \mathcal{Q}^{(P)}, Q_2 \in \mathcal{Q}^{(C)}} D_2(Q_1, Q_2) s.t. R_1(Q_1) + R_2(Q_1, Q_2) \le R_{budget}. \tag{29}$$

As is well known, one can introduce the Lagrange multiplier $\lambda$ and Lagrangian costs $J_1(Q_1) = \lambda R_1(Q_1)$ and $J_2(Q_1, Q_2) = D_2(Q_1, Q_2) + \lambda R_2(Q_1, Q_2)$ and solve, for the "correct" value of $\lambda$, $\lambda^*$, the equivalent unconstrained problem

$$\min_{\{Q_1 \in \mathcal{Q}^{(P)}, Q_2 \in \mathcal{Q}^{(C)}\}} [J_1(Q_1) + J_2(Q_1, Q_2)] \tag{30}$$

However, (30) requires an exhaustive search over all block quantizer choices for layers 1 and 2, unless we exploit the dependency constraint of the parent/children blocks, i.e., (30) implies

$$\begin{aligned}&\min_{Q_1}[J_1(Q_1) + \min_{Q_2} J_2(Q_1, Q_2)]\\&\quad = \min_{Q_1}[J_1(Q_1) + \sum_{j=1}^{N} \min_{\{q_{2,j} \in Q_j^c\}} J_{2,j}(Q_1, q_{2,j})] \qquad (31)\end{aligned}$$

where $J_{2,j}(.)$ is the Lagrangian cost for layer 2, block $j$.

Now, imposing the parent/child dependency constraint of each (coarse) parent block $i$ influencing only its four (residue) children blocks $4i-3, 4i-2, 4i-1, 4i$, we have

$$J_{2,4i-k}(Q_1, q_{2,4i-k}) = J_{2,4i-k}(q_{1,i}, q_{2,4i-k}) for k = 0, \ldots, 3. \qquad (32)$$

Using this, (31) becomes

$$\min_{Q_1}[J_1(Q_1) + \sum_{i=1}^{N/4}\sum_{k=0}^{3} \min_{q_{2,4i-k}} J_{2,4i-k}(q_{1,i}, q_{2,4i-k})] \qquad (33)$$

$$= \sum_{i=1}^{N/4} \min_{q_{1,i}}[J_{1,i}(q_{1,i}) + \sum_{k=0}^{3} \min_{q_{2,4i-k}} J_{2,4i-k}(q_{1,i}, q_{2,4i-k})] \qquad (34)$$

$$= \sum_{i=1}^{N/4} \min_{q_{1,i} \in Q_i^p} [J_{1,i}(q_{1,i}) + J_{2,i}^*(q_{1,i})], \qquad (35)$$

where $J_{2,i}^*(q_{1,i}) = \sum_{k=0}^{3} \min J_{2,4i-k}(q_{1,i}, q_{2,4i-k})$ in (35). As can be seen from (35), the optimization operation has been reduced to independent minimizations over each of the $N/4$ parent/child "sets" because of the dependency constraint.

## REFERENCES

[1] J.-Y. Huang and P. M. Schultheiss, "Block quantization of correlated gaussian random variables," *IEEE Trans. Commun.*, vol. 11, pp. 289–296, Sept. 1963.

[2] N. S. Jayant and P. Noll, *Digital coding of waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[3] A. Segall, "Bit allocation and encoding for vector sources," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 162–169, Mar. 1976.

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

[5] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, pp. 532-540, Apr. 1983.

[6] K. M. Uz, M. Vetterli, and D. LeGall, "Interpolative multiresolution coding of advanced television with compatible subchannels," *IEEE Trans. CAS Video Technology, Special Issue Signal Processing Advanced Television*, vol. 1, pp. 86–99, Mar. 1991.

[7] D. LeGall, "MPEG: a video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, pp. 46–58, Apr. 1991.

[8] K. Ramchandran, A. Ortega, K. M. Uz, and M. Vetterli, "Multiresolution joint source and channel coding for digital broadcast of HDTV," *To appear, IEEE J. Selected Areas Commun.*, 1993.

[9] K. Ramchandran, A. Ortega, and M. Vetterli, "Efficient quantization for a multiresolution HDTV source coder," in *Proc. Fifth Int. Workshop HDTV* (Tokyo), Nov. 1992, pp. 87.1–87.8, vol. II.

[10] ——, "Bit allocation for dependent quantization with applications to MPEG video coders," in *Proc. ICASSP'93*, Apr. 1993.

[11] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate–distortion sense," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 160–175, Apr. 1993.

[12] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[13] J. M. Salinas and R. L. Baker, "Laplacian pyramid encoding: Optimum rate and distortion allocations," *Proc. ICASSP*, 1989, pp. 1957–1960.

[14] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, pp. 30–44, Apr. 1991.

**Kannan Ramchandran** was born in Madras, India, in 1961. He received the B.S. degree in electrical engineering from the City College of New York in 1982 and the M.S. and Ph.D. degrees in electrical engineering from Columbia University in 1984 and 1993, respectively.

From 1982 to 1984, he worked for IBM as a microprocessor systems designer. From 1984 to 1990, he was a Member of the Technical Staff at AT&T Bell Laboratories as a telecommunications research and design engineer in the optical fiber loop carrier system and the fiber-to-the-home system. From 1990 to 1993, he was a Graduate Research Assistant at the Center for Telecommunications Research at Columbia University. Since 1993, he has been with the University of Illinois at Urbana-Champaign, where he is currently an Assistant Professor. His research interests include multiresolution and multirate signal processing, joint source and channel coding, adaptive quantization, and fast algorithms and architectures for image and video compression.

Dr. Ramchandran was the recipient of the 1993 "Elaihu I. Jury Award" at Columbia University for outstanding achievement as a graduate student.

**Antonio Ortega** (S'92) was born in Madrid, Spain, in 1965. He received the Telecommunication Engineering degree from the Universidad Politecnica de Madrid (UPM), Madrid, Spain, in 1989 and is currently working toward the Ph.D. degree at Columbia University, New York, NY.

In 1986 and 1987, he worked during the summer for Gibbs and Hill, Inc., Washington, DC. In 1988, he was a summer intern at the Optoelectronics Lab, ENST, Paris, France. During 1990, he was a research assistant at the Image Processing Group at UPM. Since January 1991, he has been a Research Assistant at the Center for Telecommunications Research, Columbia University. His work is being supported by a scholarship from the Fulbright Commission and the Spanish Ministry of Education. His research interests include digital image and video processing and communications.

**Martin Vetterli** (SM'90) received the Dipl. El.-Ing. degree from ETH Zürich, Switzerland, in 1981, the M.S. degree from Stanford University in 1982, and the Doctorat ès Science degree from EPF Lausanne, Switzerland, in 1986.

He was a Research Assistant at Stanford University and EPFL and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University, New York, NY, where he is currently Associate Professor of Electrical Engineering. Since July 1993, he has also been on the faculty of the Department of Electrical Engineering and Computer Science at the University of California, Berkeley. His research interests include wavelets, multirate signal processing, computational complexity, signal processing for telecommunications, and digital video processing and compression.

Dr. Vetterli is a member of SIAM and ACM, and of the editorial boards of *Signal Processing, Image Communication, Annals of Telecommunications Applied and Computational Harmonic Analysis*, and *The Journal of Fourier Analysis and Applications.* He received the Best Paper Award of EURASIP in 1984 for his paper on multidimensional subband coding, the Research Prize of the Brown Bovery Corporation (Switzerland) in 1986 for his thesis, and the IEEE Signal Processing Society's 1991 Senior Award for a 1989 Transactions paper with D. LeGall. He was a plenary speaker at the 1992 IEEE ICASSP and is the co-author, with J. Kovacevic, of the forthcoming book *Wavelets and Subband Coding* (Prentice-Hall, 1994).