Multirate 3-D Subband Coding of Video

David Taubman and Avideh Zakhor

Abstract—**We propose a full color video compression strategy, based on 3-D subband coding with camera pan compensation, to generate a single embedded bit stream supporting multiple decoder display formats and a wide, finely gradated range of bit rates. An experimental implementation of our algorithm produces a single bit stream, from which suitable subsets are extracted to be compatible with many decoder frame sizes and frame rates and to satisfy transmission bandwidth constraints ranging from several tens of kilobits per second to several megabits per second. Reconstructed video quality from any of these bit stream subsets is often found to exceed that obtained from an MPEG-1 implementation, operated with equivalent bit rate constraints, in both perceptual quality and mean squared error. In addition, when restricted to 2-D, the algorithm produces some of the best results available in still image compression.**

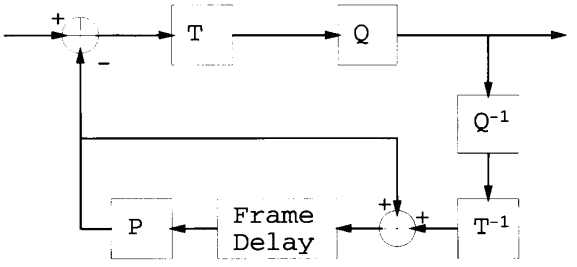

Fig. 1. Hybrid compression schematic. T denotes DCT or subband decomposition; Q denotes quantization; P denotes temporal prediction.

I. Introduction

In this work we present an efficient video compression strategy for applications in which the encoder is to operate independently of decoder display resolution and transmission bandwidth limitations. A multicast scenario, in which many decoders with differing display resolutions are serviced by a distributed network with various time-varying bandwidth constraints, provides a generic example of such applications. We identify two design goals. Firstly, the encoder must produce a single embedded bit stream suitable for a variety of different decoder display resolutions. By display resolution, we connote frame size and frame rate. We shall refer to this first of our design goals as the *multiresolution* property of the compression strategy. An encoded bit stream supporting multiresolution decoding provides some measure of bit rate scalability through the selection of different display resolutions. Our second design goal, however, is to expand this rate scalability to allow for extraction of subsets of the bit stream, covering a range of several tens of kilobits per second through several megabits per second, with a fine gradation of available data rates throughout this range. We shall refer to this, our second design goal, as the *multirate* property of the compression strategy.

The multiresolution property of the compression strategy is relevant to a wide variety of video applications. Such applications include embedded television standards to support conventional television through to HDTV and beyond, variable quality videophony, windowed video display on workstations and a variety of multimedia applications. On the other hand the ability to readily trade transmission bandwidth for quality at any point in the distribution, by selecting subsets of the embedded bit stream, is particularly relevant to storage and transmission issues. Such applications as video transmission over ATM networks and storage device contention in video-on-demand databases are examples. This multirate property is important to most video compression applications because of the inevitable storage and transmission issues involved.

Most popular video coding strategies are based on a hybrid structure in which 2-D transform or subband coding is applied to the prediction error signal from temporal prediction, usually in the form of motion compensation. In general, such strategies may be represented by the schematic of Fig. 1, in which the state of an assumed decoder is maintained within the predictive feedback loop. A number of proposals have been advanced to extend the structure of Fig. 1 to allow multiple display resolutions. Civanlar and Puri [5] propose a minimal departure from the MPEG-1 standard, in which a single predictive feedback loop maintains the state of the highest resolution decoder, while lower spatial and temporal resolutions are obtained by discarding transform coefficients and frames, respectively. This approach has also been adopted by CCITT Working Party SG XV/1 [3]. It has the difficulty, however, that the predictive feedback loop can only track the state of a decoder with one resolution, leading to prediction drift for the other decoders. Vandendorpe [14], on the other hand, advocates the application of separate predictive feedback loops to the coding of several spatial resolutions obtained by 2-D transform or subband techniques. His approach removes the prediction drift associated with a single feedback loop. For high motion scenes, however, he observes an increase of as much as 60% in overall bit rate with his multiresolution approach as compared with the conventional structure of Fig. 1. A related scheme has been proposed by Zhang and Zafar [16]. In contrast to adaptation of the hybrid structure, Bove and Lippman [2] have proposed a completely separable 3-D subband scheme for multiresolution compression.

Manuscript received April 1, 1993; revised February 20, 1994. This work was supported by Sun Microsystems, National Science Foundation PYI award MIP-9057466, and the ONR young investigator award N00014-92-J-1732.

The authors are with the University of California, Berkeley, CA 94720 USA.

IEEE Log Number 9402245.

The approach adopted in this paper is to predistort the video sequence by translating frames relative to one another so as to compensate for camera pan motion. A separable 3-D subband decomposition is then applied to the shifted frame sequence. The combination of frame translation and separable subband filtering may be viewed as a nonseparable subband decomposition with velocity sensitivity. This approach is related to that of Ohm [8], who uses motion vectors within a 3-D subband coding context. It is also related to a still image compression strategy of Taubman and Zakhor [13], in which images are predistorted by spatial reorientation prior to 2-D subband coding. By exploiting temporal correlation through temporal subband filtering, we eliminate the troublesome predictive feedback loop of Fig. 1, thereby eliminating dependence on the receiver resolution.

The compression strategy described in this paper may be viewed as an extension of results first presented by Chang and Zakhor [4]. Our main deviations from their work include a significantly improved and more widespread application of camera pan compensated temporal subband decomposition, and a more efficient exploitation of the numerous correlations available for coding of the quantized subband values. In Section II we describe our proposed approach to subband decomposition, whereby an embedded bit stream possessing the multiresolution property may be obtained. The novelty of our compression strategy, however, lies not so much in its multiresolution property as it does in its multirate attributes. The generation of an embedded multirate bit stream, by progressive coding of each of the subbands, is discussed in Section III. In Section IV we present experimental results obtained with several standard video test sequences, comparing the performance of our strategy to fixed-rate, multiresolution subband coding as well as an implementation of the fixed-rate, fixed-resolution MPEG-1 standard. In Section IV we also present results in still image compression, obtained by restricting the 3-D compression algorithm to 2-D.

II. Subband Structure

As mentioned in Section I, our first design goal is to generate an embedded bit stream with the multiresolution property, whereby subsets of the bit stream may be independently decoded for different display frame sizes and frame rates. We propose the well-known technique of separable subband decomposition to achieve this goal. Separable spatial subband techniques were first applied to images by Woods [15]. Spatial subband coding allows for exploitation of both the statistical and phsycho-visual properties of the subbands for efficient compression, while at the same time providing for reconstruction at a variety of spatial resolutions, i.e., frame sizes, from subsets of the bit stream. Separable extension of spatial subband decomposition along the temporal dimension of a video sequence allows for reconstruction at a variety of temporal resolutions, i.e., frame rates, from subsets of the bit stream, while permitting additional coding gain. In the presence of camera and scene motion, however, separable 3-D subband coding can suffer from two significant difficulties. The high spatial frequency subbands typically contain spatial edge information. When these edges are in motion, temporal filtering causes this edge information to be smeared and can lead to a decrease in coding efficiency. The second, related difficulty is the subjective degradation introduced by the smearing of background features during reconstruction at a reduced frame rate. When a significant proportion of scene motion results from camera pan, both of these difficulties can be largely overcome by predistorting the video sequence so as to compensate for camera pan motion. In Section II-A, we discuss our approach to separable subband decomposition in 2-D and 3-D. Camera pan compensation is discussed in Section II-B.

A. Spatiotemporal Subband Structure

Although an enormous variety of subband structures may be obtained by combining separable spatial and temporal filtering operations, the approach adopted here is to consider a classical spatial decomposition hierarchy and extend it to 3-D by applying simple temporal decomposition structures to each of the resulting spatial subbands. The complete structure is based on separable and recursive application of the familiar 1-D two-channel subband decomposition, whose subbands correspond to low and high frequency passbands. Our particular selection of 1-D subband filters is indicated in Section IV-A. Separable application of the two-channel system, in the horizontal and vertical spatial dimensions, leads to a four-channel decomposition, with subbands denoted S—LL, S—LH, S—HL and S—HH. In this notation S— signifies spatial filtering, with the first L or H respectively referring to a low or high frequency passband in the horizontal direction, and the second L or H referring to a low or high frequency vertical passband. The spatial decomposition may be extended by iteratively decomposing the S-LL band into a further four subbands. As an example, Fig. 2 illustrates a useful structure for color image compression. In this case, the luminance component is applied to a five-level spatial decomposition hierarchy, whereas the horizontally and vertically subsampled chrominance components are decomposed in a four-level hierarchy.

To extend such spatial hierarchies to 3-D we consider structures in which each spatial subband may be applied to a two-channel temporal decomposition, yielding low and high temporal frequency subbands, denoted T—L and T—H, respectively. The T—L subband may be further divided into temporal subbands, so that each spatial subband may be decomposed in a temporal subband hierarchy. Fig. 3 provides an example of a 3-D subband structure conforming to this model. The four spatial and three temporal decomposition levels of Fig. 3 are found to provide broadly optimal compression performance in our experimental work with several standard video test sequences. The 49 luminance and 37 chrominance subbands of this subband structure support the numerous decoder display resolutions listed in Table I. A decoder need only decode that portion of the bit stream representing the subbands associated with its target display resolution. The decoder for a monochrome display of frame size 176 × 120 at 15 frames per second, for example, need only receive and decode code bits for subbands Y0–Y11, Y13–Y15, Y17–Y19, Y21–Y23, Y25–Y27, Y29–Y31 and Y33–Y35 of Fig. 3.

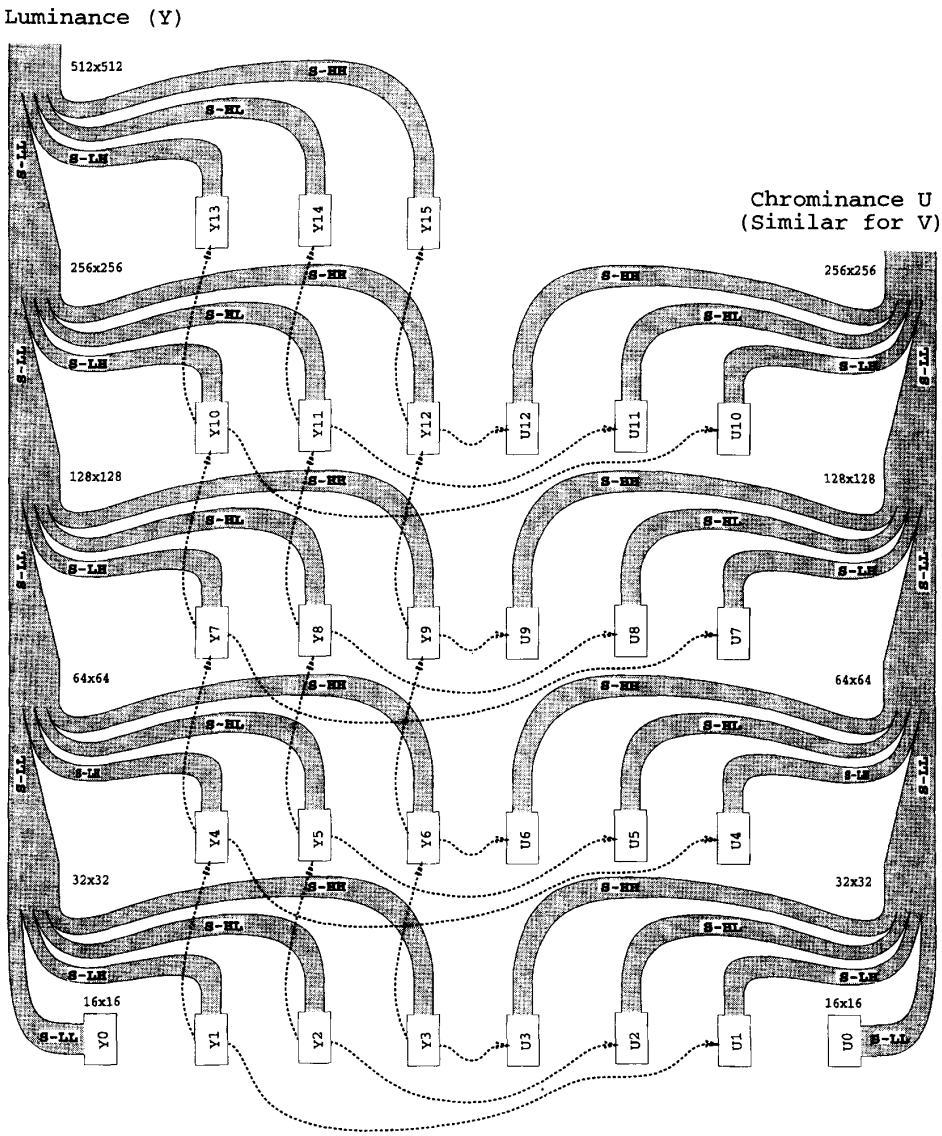

Fig. 2. Spatial subband structure. S—LL denotes spatial horizontally and vertically low pass subband; S—LH denotes spatial horizontally low pass and vertically high pass subband, etc. Dotted lines indicate relationships selected for intersubband correlation, as discussed in Section III-C-2.

B. Compensation for Camera Pan

A separable 3-D subband system is clearly best suited to video sequences for which the camera is held still. In this case only scene motion contributes information to the high temporal frequency subbands. The purpose of camera pan compensation is to predistort the video sequence so as to present the separable subband system with a modified sequence in which camera pan motion is eliminated. At the decoder this pan compensation must be inverted. Fig. 4 is helpful in understanding this approach. The figure portrays four frames of a hypothetical video sequence, with no scene motion, in which the camera pans to the right at a constant rate of one pixel per frame. Only one spatial dimension is shown, for clarity. In this case, where the camera pans by an integral number of pixels per frame, camera pan compensation consists only in relabeling the pixel indices of each of the frames. In particular, after relabeling, the pixel indices associated with frame 0 run from 0 to 12, those of frame 1 run from 1 to 13 and so on. This relabeling process does not involve any modification of pixel values and, as such, has no impact on the computational demands of the encoder or decoder, however it has a profound effect on the compression performance of the algorithm. For the camera pan compensation approach to be effective, all spatial and temporal subband filtering and subsampling operations must be performed with reference to these relabeled indices. In the spatial dimensions this *pixel*

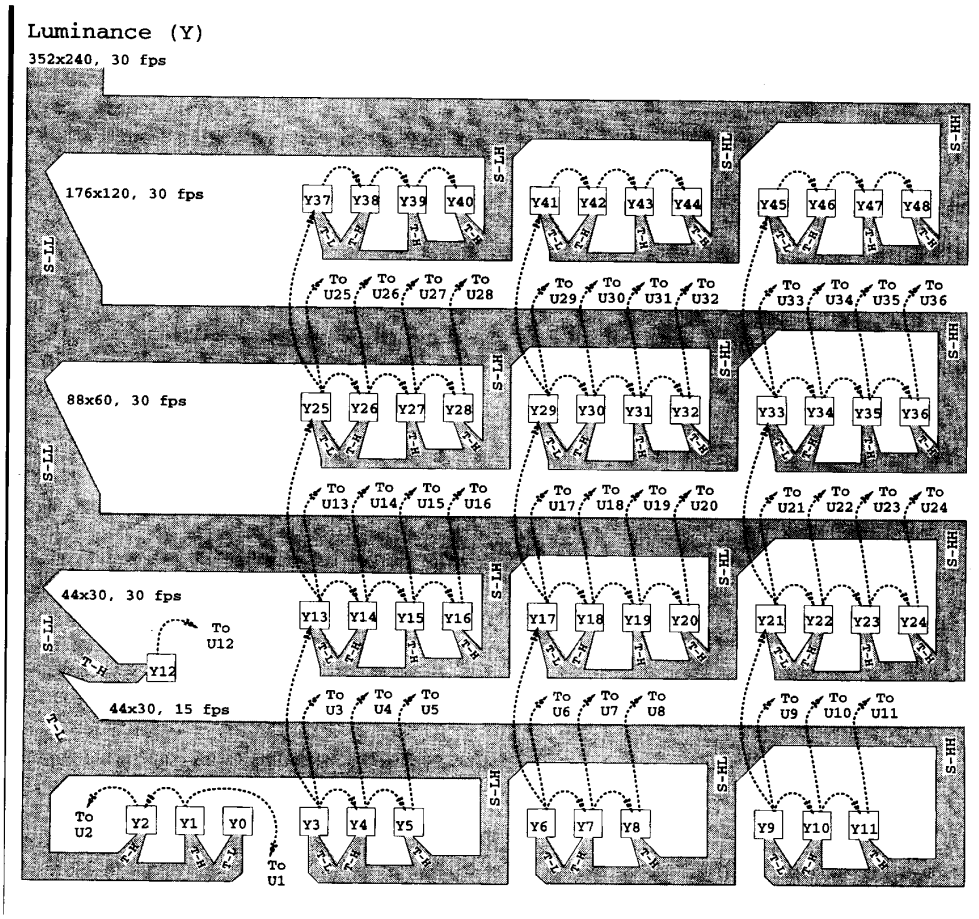

Fig. 3. Spatiotemporal subband structure. S—LL, S—LH, etc., as in Fig. 2. T—L and T—H denote temporal low and high pass subbands, respectively. Only luminance component is explicitly shown. U and V chrominance components are subsampled by two, horizontally andvertically and so their subband structures are missing subbands $37 \cdots 48$. Dotted lines indicate relationships selected for intersubband correlation, as discussed in Section III-C-2.

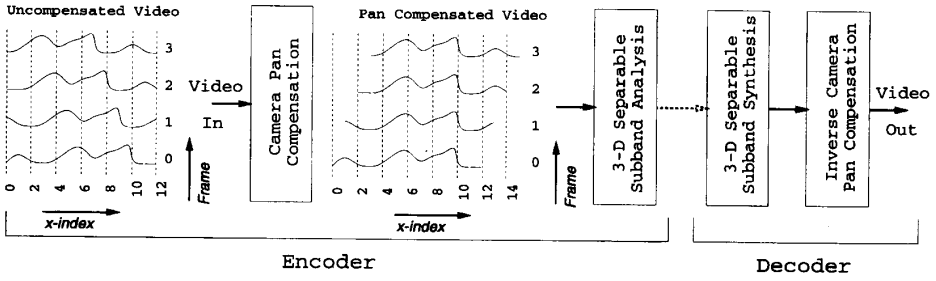

Fig. 4. Separable subband system with camera pan compensation. For clarity, only one spatial dimension is shown. Operation of camera pan model is demonstrated with four frames of a hypothetical sequence, in which camera pans to the right at a rate of one pixel per frame.

index discipline means that a spatial subsampling operation must discard the odd indexed pixels, retaining those with even index, regardless of whether the first pixel in a frame has even index, as in frames 0 and 2 of Fig. 4, or odd index, as in frames 1 and 3. In the temporal dimension, the pixel index discipline means that temporal filtering is applied to pixels with identical spatial indices in successive frames.

An important consequence of camera pan compensation is that pixels with spatial index $\vec{a}$ may exist in some but not all frames of the video sequence. We propose a modification of the well-known Haar wavelet to permit efficient temporal subband decomposition in the face of such temporal discontinuities. In particular, suppose that pixels having spatial index $\vec{a}$ exist for frames $n \in [N_1, N_2]$, but not for frames $n = N_1 - 1$ or

TABLE I
DECODER DISPLAY FORMATS SUPPORTED BY SUBBAND STRUCTURE OF FIG. 3, ASSUMING CHROMINANCE IS ALWAYS SUBSAMPLED BY TWO, HORIZONTALLY AND VERTICALLY, RELATIVE TO LUMINANCE

Frame Format	Available Frame Rates (fps)
352 × 240 color	30, 15, 7.5, 3.75
352 × 240 monochrome	30, 15, 7.5, 3.75
176 × 120 color	30, 15, 7.5, 3.75
176 × 120 monochrome	30, 15, 7.5, 3.75
88 × 60 color	30, 15, 7.5, 3.75
88 × 60 monochrome	30, 15, 7.5, 3.75
44 × 30 color	15, 7.5, 3.75
44 × 30 monochrome	30, 15, 7.5, 3.75
22 × 15 monochrome	15, 7.5, 3.75

$n = N_2 + 1$. Let $x_{\vec{a}}[n]$ denote the sequence consisting of these pixel values.[1] We define the low pass temporal subband samples, $y_{\vec{a}}[2n]$, and high pass temporal subband samples, $y_{\vec{a}}[2n+1]$, by

$$y_{\vec{a}}[2n] = \begin{cases} \frac{x_{\vec{a}}[2n]+x_{\vec{a}}[2n+1]}{\sqrt{2}} & \text{if } 2n \in [N_1, N_2) \\ \frac{2x_{\vec{a}}[2n]}{\sqrt{2}} & \text{if } 2n = N_2 \end{cases}$$

and

$$y_{\vec{a}}[2n+1] = \begin{cases} \frac{x_{\vec{a}}[2n+1]-x_{\vec{a}}[2n]}{\sqrt{2}} & \text{if } 2n \in [N_1, N_2) \\ \frac{x_{\vec{a}}[2n+1]-x_{\vec{a}}[2n+2]}{\sqrt{2}} & \text{if } 2n+1 = N_1 < N_2 \\ x_{\vec{a}}[2n+1] & \text{if } 2n+1 = N_1 = N_2 \end{cases}$$

from which the original sequence, $x_{\vec{a}}[n]$, is readily recovered. We have extended this technique to allow for temporal filtering with orthogonal 4-tap filters whose polyphase analysis matrix [7] is paraunitary. These filters can give much better spectral localization than the 2-tap filters described above, however their implementation requires more memory and we have found them to offer little if any coding or perceptual gain.

An unfortunate consequence of temporal subband filtering is that it introduces latency between picture encoding and decoding. This poses a problem only for interactive applications where it is usually reasonable to assume that the camera is stationary. In this case, with L levels of temporal decomposition, the total latency may be shown to be $\lesssim 2^{L+1}$ frames. So, at 30 frames per second, two temporal levels would give a total latency of approximately $\frac{1}{4}$ second.

As mentioned, when the camera pans by an integral number of pixels per frame, camera pan compensation is merely a relabeling of the pixel indices, and hence is effected by the pixel index discipline discussed above. The success of camera pan compensation, however, turns out to be highly dependent on the ability to achieve subpixel accuracy. Fortunately, a video sequence in which the camera pans by a nonintegral number of pixels per frame may be converted into one in which the camera pans by an integral number of pixels per frame by first shifting each frame by a maximum of $\pm\frac{1}{2}$ a pixel in the horizontal and vertical directions. For example, suppose that the pan associated with the hypothetical video sequence of Fig. 4 were $1\frac{1}{4}$ pixels per frame. Then by shifting frame 1 by $\frac{1}{4}$ pixel to the right, frame 2 by $\frac{1}{2}$ pixel to the right and frame 3 by $\frac{1}{4}$ pixel to the left, the shifted sequence exhibits a camera pan of 1 pixel per frame from frames 0 to 2 and 2 pixels per frame from frame 2 to frame 3. For subpixel accuracy, then, camera pan compensation requires interpolative shifting by at most half a pixel in each direction, together with pixel index relabeling. Moreover, the decoder must be able to invert this interpolative shift, as indicated by the "inverse camera pan compensation" block of Fig. 4. While pixel relabeling is a trivially invertible process, interpolative shifting is not. This same problem arises in [13], in which still images are invertibly reoriented so as to improve the compression performance of separable spatial subband coding. In that work, an interpolative filter is developed which corresponds to a linear shift invariant filter with all-pass magnitude response, except near frame boundaries where shift invariance is lost. The same approach is adopted here.

The camera pan parameters themselves may either be obtained as direct inputs from some sensor on the video camera, or derived by matching successive video frames. Out of necessity we adopt the latter approach. A coarse estimate of the camera pan is obtained by first replacing each pixel in each frame either by zero or by the local horizontal or vertical gradient, if the magnitude of this gradient is a local optimum, and then cross correlating consecutive frames, using the fast Fourier transform (FFT) for efficiency. This coarse estimate is refined by using telescopic block matching with 16 × 16 blocks and bilinear interpolation for subpixel resolution. The camera pan is taken to be the motion vector which gives the best match for the largest number of these blocks. This algorithm is found to be effective even with difficult sequences such as the 'football' sequence, where camera pan is often obscured by scene motion. Moreover, the algorithm's numerical complexity is dominated by the $\mathcal{O}(n\log(n))$ FFT and inverse FFT operations, as opposed to conventional block matching which has $\mathcal{O}(n^2)$ complexity.

III. MULTIRATE CODING OF SUBBANDS

In this section we discuss our second design goal, that of multirate coding. Although the subband structure presented in Section II-A admits some degree of control over bit rate through the selection of various display resolutions, the multirate coding techniques discussed in this section greatly improve the range and granularity of such rate control. This is achieved by progressive quantization and coding of each subband in a sequence of N layers, representing progressively finer quantization step sizes. In this regard, our approach bears a significant resemblance to the embedded image coder described by Shapiro [12].[2] We refer to a set of N quantizers, $\mathcal{Q}_1, \cdots, \mathcal{Q}_N$ and N quantization layers $\mathcal{L}_1, \cdots, \mathcal{L}_N$. Each quantizer is understood as operating on the subband samples to produce a sequence of symbols. The symbols for quantizer $\mathcal{Q}_1$

[1] The 'pixels' referred to here are usually themselves spatial subband samples.

[2] We note, however, that our multirate coding techniques differ significantly from those in [12]. In particular, our approach endows the bit stream with both the multirate and the multiresolution property simultaneously.

TABLE II
STATISTICAL DEPENDENCIES EXPLOITED IN CODING SUBBAND SAMPLES

Correlation	PCM	DPCM/CR	Zero Coding
Interquantization layer	√	√	√
Spatial	×	√	√
Temporal	×	√	×
Intersubband	×	×	√

are encoded into layer $\mathcal{L}_1$, while the information necessary to recover the symbols for quantizer $\mathcal{Q}_n$, given that the symbols for quantizers $\mathcal{Q}_1, \cdots, \mathcal{Q}_{n-1}$ are already known, is encoded into layer $\mathcal{L}_n$. In this way the decoder is able to recover the subband samples, as quantized by any of the quantizers, $\mathcal{Q}_n$, by decoding layers $\mathcal{L}_1, \cdots, \mathcal{L}_n$ only. A reasonable goal for such a layered coding scheme is that the total number of bits required to encode layers $\mathcal{L}_1, \cdots, \mathcal{L}_n$ be approximately the same as the number of bits required to encode the output of quantizer $\mathcal{Q}_n$ alone. This means that we do not sacrifice coding efficiency in obtaining the multirate property. We refer to this multilayer efficiency goal repeatedly in the remainder of this section.

In order to achieve such a goal it is necessary to exploit statistical dependencies between the quantization layers. In addition, numerous other dependencies exist, between subbands, between spatially adjacent subband samples and between temporally adjacent subband samples. It is not necessarily efficient nor feasible to make use of all of these dependencies in coding the subband samples. A summary of those dependencies exploited by the various coding techniques discussed in this paper appears in Table II. In Section III-A we discuss layered PCM coding, the technique applied to all high frequency subbands. In Section III-B we discuss layered DPCM as well as layered conditional replenishment (CR). These techniques are applied only to the lowest frequency subband in the spatiotemporal hierarchy. Enhanced coding efficiency can usually be obtained by devoting special attention to the zero symbol, for which layered zero coding techniques are presented in Section III-C.

A. Layered PCM

Throughout this section $x[i]$ denotes an observed sequence of subband samples, each element of which is an outcome of a random variable, X. Each quantizer, $\mathcal{Q}_n$, may be characterized by a set of disjoint quantization intervals $\mathcal{I}_{n,k_1}, \mathcal{I}_{n,k_2}, \cdots$. The PCM encoder corresponding to $\mathcal{Q}_n$ emits a symbol, k, whenever the subband sample, x, lies in $\mathcal{I}_{n,k}$. We indicate this quantization function by

$$\mathcal{Q}_n(x) = k, \text{ for } x \in \mathcal{I}_{n,k}.$$

We begin our discussion by imposing the following important condition on the set of quantizers, $\mathcal{Q}_1, \cdots, \mathcal{Q}_N$.

$$\mathbf{P}\big(X \in \mathcal{I}_{n,\mathcal{Q}_n(x[i])} \backslash \mathcal{I}_{n-1,\mathcal{Q}_{n-1}(x[i])}\big) = 0, \quad \forall x[i],\ \forall\ n \geq 2 \tag{1}$$

where $\mathbf{P}(\cdot)$ refers to the probability of an event and $\backslash$ is the set subtraction operation. Equation (1) states that every fine quantization interval, $\mathcal{I}_{n,\mathcal{Q}_n(x[i])}$, is almost entirely contained in the corresponding coarse quantization interval, $\mathcal{I}_{n,\mathcal{Q}_{n-1}(x[i])}$. The probability that X lies in any part of $\mathcal{I}_{n,\mathcal{Q}_n(x[i])}$ not contained in $\mathcal{I}_{n-1,\mathcal{Q}_{n-1}(x[i])}$ is zero. For all practical purposes, it is sufficient to think of (1) as the requirement that every quantization interval of $\mathcal{Q}_n$ is contained in some quantization interval of $\mathcal{Q}_{n-1}$.

In this section we first prove that our multilayer coding efficiency goal can be achieved if and only if (1) is satisfied. We then show how this condition can be applied to the design of a useful set of quantizers.

Assuming that each subband sample, $x[i]$, is a statistically independent outcome of X, the observed sequence of quantization symbols, $\mathcal{Q}_n(x[i])$, has an amount of information, $\mathcal{C}(\mathcal{Q}_n(x[i]))$, given by

$$\mathcal{C}(\mathcal{Q}_n(x[i])) = -\sum_i \log_2 \mathbf{P}\big(X \in \mathcal{I}_{n,\mathcal{Q}_n(x[i])}\big) \tag{2}$$

The notation, $\mathcal{C}$, is chosen because $\mathcal{C}(\mathcal{Q}_n(x[i]))$ is also the information theoretical lower bound on the number of code bits required to encode the sequence, $\mathcal{Q}_n(x[i])$. Moreover, this lower bound may, effectively, be achieved by using arithmetic coding [10]. Note that the minimum average number of bits required to code each symbol, $\mathcal{Q}_n(x[i])$, is the entropy, $\mathcal{H}(\mathcal{Q}_n(X))$, of the random variable, $\mathcal{Q}_n(X)$, whereas $\mathcal{C}(\mathcal{Q}_n(x[i]))$ is the minimum number of bits required to encode the *entire observed* symbol sequence.

For $n \geq 2$, layer $\mathcal{L}_n$ must encode the symbols $\mathcal{Q}_n(x[i])$, given that layers $\mathcal{L}_1, \cdots, \mathcal{L}_{n-1}$ and hence $\mathcal{Q}_{n-1}(x[i])$ are known. The minimum number of bits required to encode layer $\mathcal{L}_n$ is given by

$$\begin{aligned}
&\mathcal{C}(\mathcal{L}_n) \\
&\quad = -\sum_i \log_2 \mathbf{P}\big(X \in \mathcal{I}_{n,\mathcal{Q}_n(x[i])} \mid X \in \mathcal{I}_{n-1,\mathcal{Q}_{n-1}(x[i])}\big) \qquad (3)\\
&\quad = -\sum_i \log_2 \frac{\mathbf{P}\big(X \in \mathcal{I}_{n,\mathcal{Q}_n(x[i])} \cap \mathcal{I}_{n-1,\mathcal{Q}_{n-1}(x[i])}\big)}{\mathbf{P}\big(X \in \mathcal{I}_{n-1,\mathcal{Q}_{n-1}(x[i])}\big)} \\
&\quad = -\sum_i \log_2 \Big[\mathbf{P}\big(X \in \mathcal{I}_{n,\mathcal{Q}_n(x[i])}\big) \\
&\qquad\qquad - \mathbf{P}\big(X \in \mathcal{I}_{n,\mathcal{Q}_n(x[i])} \backslash \mathcal{I}_{n,\mathcal{Q}_{n-1}(x[i])}\big)\Big] \\
&\qquad + \sum_i \log_2 \mathbf{P}\big(X \in \mathcal{I}_{n-1,\mathcal{Q}_{n-1}(x[i])}\big) \\
&\quad \geq \mathcal{C}(\mathcal{Q}_n(x[i])) - \mathcal{C}(\mathcal{Q}_{n-1}(x[i])) \qquad (4)
\end{aligned}$$

where equality holds if and only if $\mathbf{P}\big(X \in \mathcal{I}_{n,\mathcal{Q}_n(x[i])} \backslash \mathcal{I}_{n-1,\mathcal{Q}_{n-1}(x[i])}\big) = 0$ – i.e., if and only if (1) is satisfied. Noting that layer $\mathcal{L}_1$ simply encodes the sequence $\mathcal{Q}_1(x[i])$, requiring $\mathcal{C}(\mathcal{Q}_1(x[i]))$ bits, (4) leads, by induction on n, to the result

$$\sum_{m=1}^{n} \mathcal{C}(\mathcal{L}_m) \geq \mathcal{C}(\mathcal{Q}_n(x[i])) \tag{5}$$

with equality if and only if (1) is satisfied. Equation (5) states that our efficiency goal is satisfied for layered PCM coding, provided arithmetic coding is employed so that the minimum number of bits, $\mathcal{C}(\mathcal{L}_n)$, is achieved in coding each quantization

layer, $\mathcal{L}_n$, and provided (1) is satisfied. In this case the number of bits required to encode the first n quantization layers, from which the quantization symbols, $\mathcal{Q}_n(x[i])$, may be recovered, is the same as the number of bits required to encode the quantization symbols $\mathcal{Q}_n(x[i])$ directly. This means that we need not sacrifice coding efficiency to obtain the multirate property.

Before turning our attention to more pragmatic issues of layered PCM coding, we provide an alternative interpretation of (1), in order to impart an intuitive understanding of its significance. Equation (1) essentially states that every quantization interval, $\mathcal{I}_{n,k}$, of $\mathcal{Q}_n$ is a subset of some quantization interval, $\mathcal{I}_{n-1,k'}$ of $\mathcal{Q}_{n-1}$. This means that, given the sequence $\mathcal{Q}_n(x[i])$, the sequences $\mathcal{Q}_{n-1}(x[i])$, $\mathcal{Q}_{n-2}(x[i]), \cdots$ can all be deduced. Thus the n quantization layers, $\mathcal{L}_1, \cdots, \mathcal{L}_n$, convey no more information about the subband samples, $x[i]$, than the single quantizer output, $\mathcal{Q}_n(x[i])$. It follows that (1) may be interpreted as the condition that minimizes the information represented by the collection of quantization layers, $\mathcal{L}_1, \cdots, \mathcal{L}_n$, used to encode $\mathcal{Q}_n(x[i])$.

We choose to use arithmetic coding in our multirate coder because it is able to approach the information theoretical lower bounds, $\mathcal{C}(\mathcal{L}_n)$, arbitrarily closely. Huffman coding is not a viable alternative here because it cannot realize bit rates of less than one bit per symbol[3]; one bit per symbol being an upper bound on $\mathcal{C}(\mathcal{L}_n)$ for $n \geq 2$ if the quantization step size is chosen so as to be halved in each successive layer. In order to achieve this minimum number of code bits, $\mathcal{C}(\mathcal{L}_n)$, as given by (3), the arithmetic coder must have access to the probabilities $\mathbf{P}(X \in \mathcal{I}_{n,k} \mid X \in \mathcal{I}_{n-1,k'})$ for all k, k' and n. Equation (1), however, tells us that every interval, $\mathcal{I}_{n,k}$ is the subset of some other interval $\mathcal{I}_{n-1,k'}$ whence it may be shown that the number of nontrivial probabilities—i.e., not 0 or 1—, to which the arithmetic coder must have access, cannot exceed twice the number of highest resolution quantization intervals, $\mathcal{I}_{N,k}$. This means that the total memory requirement of the probability tables required to encode all N quantization layers is no more than twice the memory requirement of the probability table for arithmetic coding of the single precision sequence $\mathcal{Q}_N(x[i])$. In our judgement, this is not a great price to pay for the multirate property.

We turn our attention now to design of the quantizers. From a mean squared error perspective, the minimum entropy is approximately achieved for a given distortion by uniform quantization (see p. 154 of [6].[4] In addition, however, we observe that our layered PCM coder is to be applied to the high frequency subbands, where the sample values, x, are frequently small. In order to increase the efficiency of zero coding techniques, such as those discussed in Section III-C, it is typical to use a larger quantization interval around $x = 0$; this interval is known as a dead zone. For quantizer $\mathcal{Q}_n$, then, we identify a quantization step size, Δ_n, and a dead zone threshold, Θ_n, such that

$$\mathcal{I}_{n,k} = \begin{cases} (-\Theta_n, \Theta_n) & \text{if } k = 0 \\ [\Theta_n + (k-1)\Delta_n, \Theta_n + k\Delta_n) & \text{if } k > 0 \\ (-\Theta_n + k\Delta_n, -\Theta_n + (k+1)\Delta_n] & \text{if } k < 0 \end{cases}.$$

For a set of N such uniform quantizers with dead zones, it can be shown that (1) holds if and only if the following two conditions are satisfied

$$\Delta_{n-1} = K_n \Delta_n \tag{6}$$

$$\Theta_n = \Theta_{n-1} - K'_n \Delta_n \tag{7}$$

where $K_n > 0$ and $K'_n \geq 0$ are arbitrary integers. Equation (6) indicates that the quantization step size must be divided by an integral factor from layer to layer. In practice we choose the smallest useful factor, $K_n = 2$, in which case *each successive quantization layer doubles the precision to which subband sample values are quantized.* Equation (7) arises from the fact that the quantizer dead zone must be centred about 0. For consistency and ease of implementation, we would like to halve not only the quantization step size, Δ_n, but also the dead zone threshold, Θ_n, in each successive quantization layer. One way to arrange this is to set $\Theta_n = \Delta_n$, in which case (7) is satisfied with $K'_n = 1$. This means that *each quantizer has a dead zone twice as large as the other quantization intervals.*

B. Layered DPCM/CR

In this section we discuss the application of multilayer quantization and coding techniques to the predictive coding strategies of delta pulse code modulation (DPCM) and conditional replenishment (CR). In our algorithm, these strategies are applied to the subband of lowest temporal and spatial frequency. As in the PCM case, our goal is to avoid sacrificing coding efficiency in order to obtain the multirate property. We show that the coding efficiency for layered DPCM and CR can significantly exceed that obtained by simple bit-planing techniques however, unlike PCM coding, the efficiency goal cannot be realized for layered DPCM/CR coding.

For predictive coding we identify a sequence of current sample values, $x[i]$, to be coded, and a sequence of reference sample values, $\tilde{x}[i]$, whose quantized values are available to the decoder and are to be used in predicting $x[i]$. As in Section III-A, we view the elements of these sequences as particular outcomes of random variables, X and $\tilde{X}$, respectively. The assumption of predictive coding is that X and $\tilde{X}$ are not independent. The sequence, $x[i]$, is obtained by scanning each frame of the subband in a zig-zag fashion, with the first row being scanned from left to right, the second row from right to left, and so on. For DPCM the reference sequence is simply given by $\tilde{x}[i] = x[i-1]$, with $x[0]$ coded independently. Conditional Replenishment is an interframe coding technique, in which $\tilde{x}[i]$ is a sample with the same spatial indices—see Section II-B—as $x[i]$, but in the previous frame. A consequence of index relabeling during camera pan compensation is that such a reference pixel may not exist for every sample $x[i]$. In these cases the reference pixels are given by $\tilde{x}[i] = x[i-1]$ as for DPCM. In our proposed algorithm the encoder assesses the correlation between sequences $x[i]$

[3] Although vector quantization, in combination with Huffman coding, would allow for rates of less than one bit per sample, it can be shown that the codebook memory requirements for our application are prohibitively large.

[4] This comment is only strictly correct at relatively high bit rates.

and $\tilde{x}[i]$ for both the DPCM and CR methods and the method resulting in the highest correlation is selected. This approach allows the optimal method to be selected for both high and low motion scenes.

Predictive coders such as DPCM and CR require the quantizer to be included inside a feedback loop which keeps track of the quantized reference values available at the decoder.[5] As mentioned in Section I, such feedback loops work against multirate schemes, in which there is no fixed decoder state. A single feedback loop leads to problems of prediction drift for all quantizers of lower precision than that used to generate the feedback state, and reduced coding efficiency for all quantizers of higher precision. On the other hand, the application of a separate feedback loop to each of the N quantizers leads to a greater degree of independence between the N layers of DPCM/CR difference symbols, which adversely affects coding efficiency.

To avoid these difficulties we constrain the DPCM/CR quantizers to be uniform. A uniform quantization step size, Δ_n, and an initial decoder state, α_n may be associated with each of the DPCM/CR feedback loops. In the appendix we demonstrate that the sequence of quantized DPCM/CR symbols, denoted by the integer sequence $\delta\mathcal{Q}_n[i]$, is given by

$$\delta\mathcal{Q}_n[i] = \mathcal{Q}_n(x[i]) - \mathcal{Q}_n(\tilde{x}[i]) \tag{8}$$

where $\mathcal{Q}_n(x)$ is the uniform quantization operation defined by

$$\mathcal{Q}_n(x) \triangleq \left\langle \frac{x - \alpha_n}{\Delta_n} \right\rangle \tag{9}$$

in which $\langle\cdot\rangle$ denotes rounding to the nearest integer. Moreover, the corresponding sequence of DPCM/CR decoder outputs is simply $\alpha_n + \Delta_n\mathcal{Q}_n(x[i])$, the uniform quantization of sequence, $x[i]$. The significance of (8) is that, in this special case of uniform quantization, the conventional DPCM/CR feedback loop is not necessary; the quantized DPCM/CR symbols may be obtained by quantizing $x[i]$ and $\tilde{x}[i]$ independently and then subtracting the quantized values.[6] It may be shown that this 'unwrapping' of the DPCM/CR feedback loop is possible only with uniform quantization. Equation (8) is fundamental to the layered DPCM/CR strategy we now develop and analyse, in section III-B-1. Then, in Section III-B-2 we derive the compression performance of this layered coding technique, assuming an exponential model for the distribution of DPCM/CR difference symbols.

1) Analysis of Layered DPCM/CR With Uniform Quantization: As mentioned above, the output sequences from a layered DPCM/CR decoder with uniform quantization are simply $\alpha_1+\Delta_1\mathcal{Q}_1(x[i]),\cdots,\alpha_N+\Delta_N\mathcal{Q}_N(x[i])$, exactly as in the PCM case. This connection with the PCM case discussed in Section III-A provides valuable insight into the design of layered DPCM/CR quantizers. As discussed in Section III-A, the amount of information about $x[i]$ conveyed by $\mathcal{Q}_n(x[i])$ is a superset of that conveyed by $\mathcal{Q}_{n-1}(x[i])$ if and only if (1) is satisfied. Thus, even in the DPCM/CR case discussed here, (1) is important in limiting the amount of information contained in the collection of decoded sequences, $\alpha_1 + \Delta_1\mathcal{Q}_1(x[i]),\cdots,\alpha_N + \Delta_N\mathcal{Q}_N(x[i])$. Only then can we hope to limit the number of bits required to encode the layered DPCM/CR symbols, $\delta\mathcal{Q}_n[i]$, which convey this information.

[5] To avoid confusion, we stress that the reference sequence, $\tilde{x}[i]$, is known precisely only at the transmitter. The DPCM/CR receiver only has an approximation to $\tilde{x}[i]$ as a result of quantization.

[6] Equation (8) is not just a useful mathematical trick. This is actually how we implement the uniform DPCM/CR quantization.

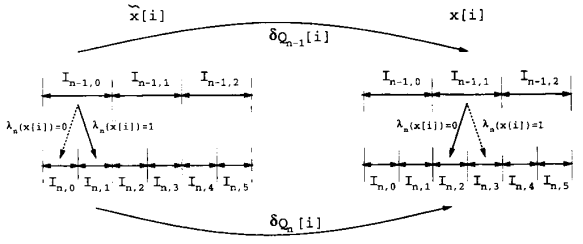

Fig. 5. Illustration of layered DPCM/CR coding strategy, with $K_n = 2$.

The quantization intervals corresponding to the uniform quantization operation, $\mathcal{Q}_n(\cdot)$, defined in (9), are given by

$$\mathcal{I}_{n,k} = \left[\alpha_n + \left(k - \frac{1}{2}\right)\Delta_n, \alpha_n + \left(k + \frac{1}{2}\right)\Delta_n\right)$$

whence α_n is rightly seen as a parameter of the quantizer $\mathcal{Q}_n$. Equation (1) may be satisfied by making the selections

$$\alpha_n = \alpha_{n-1} + \frac{1}{2}(\Delta_n - \Delta_{n-1}), \qquad \Delta_n = \frac{\Delta_{n-1}}{K_n} \tag{10}$$

where K_n is an arbitrary integer.

Fig. 5 provides a framework for understanding the operation of the layered DPCM/CR strategy to be developed. In this figure the parameter, K_n, of (10) has been set to $K_n = 2$. Equation (10) guarantees that the quantization intervals, $\mathcal{I}_{n,2k}$ and $\mathcal{I}_{n,2k+1}$, are subsets of $\mathcal{I}_{n-1,k}$, as shown in the figure, thus satisfying (1). The example of Fig. 5 portrays difference symbols of $\delta\mathcal{Q}_{n-1}[i] = 1$ and $\delta\mathcal{Q}_n[i] = 1$ for a particular sequence index, i. It is helpful to define the integer valued refinement function, λ_n, by

$$\lambda_n(x) \triangleq \mathcal{Q}_n(x) - K_n\mathcal{Q}_{n-1}(x). \tag{11}$$

By substituting (9) and (10) into (11), it may be shown that $\lambda_n(x)$ satisfies

$$0 \le \lambda_n(x) < K_n, \forall x. \tag{12}$$

Fig. 5 depicts the two possible values of the refinement function, λ_n, when $K_n = 2$.

To understand the purpose of the refinement function, λ_n, it is important to be aware of the order in which information is to be encoded in layered DPCM/CR. First, quantization layer $\mathcal{L}_1$ encodes the entire sequence of difference symbols, $\delta\mathcal{Q}_1[i]$. Layer $\mathcal{L}_2$ then updates the information contained in layer $\mathcal{L}_1$ so that the entire sequence of difference symbols, $\delta\mathcal{Q}_2[i]$, may be decoded, and so forth. The order in which each layer, $\mathcal{L}_n$, is encoded guarantees that, $\mathcal{Q}_{n-1}(\tilde{x}[i])$, $\mathcal{Q}_n(\tilde{x}[i])$ and $\mathcal{Q}_{n-1}(x[i])$ are all available at the decoder before $\delta\mathcal{Q}_n[i]$ is decoded. The encoder can use this information. In particular, we can use conditional arithmetic coding, as for layered PCM, to encode $\delta\mathcal{Q}_n[i]$, conditioned on the reference

refinement value, $\lambda_n(\tilde{x}[i])$, as well as on $\delta\mathcal{Q}_{n-1}[i]$. In fact, if we assume that the sample values, $x[i]$, are uniformly distributed and that each quantized difference symbol, $\delta\mathcal{Q}_n[i]$, is an independent outcome of some random variable, then the values of $\delta\mathcal{Q}_{n-1}[i]$and $\lambda_n(\tilde{x}[i])$ actually provide the optimal conditioning context for coding $\delta\mathcal{Q}_n[i]$. These assumptions are approximately valid for the low frequency subband samples targeted by our layered DPCM/CR coding approach. In the remainder of this section we analyse the efficiency of such conditional coding for layered DPCM/CR.

We assume that each element of the sequence of quantized differences, $\delta\mathcal{Q}_n[i] = \mathcal{Q}_n(x[i]) - \mathcal{Q}_n(\tilde{x}[i])$, is an independent outcome of the random variable $\mathcal{Q}_n(X) - \mathcal{Q}_n(\tilde{X})$. The minimum number of bits required to encode this sequence is then

$$\mathcal{C}(\delta\mathcal{Q}_n[i]) = -\sum_i \log_2 \mathbf{P}\Big(\mathcal{Q}_n(X) - \mathcal{Q}_n(\tilde{X}) = \delta\mathcal{Q}_n[i]\Big).$$

Also, the number of bits associated with our conditional coding approach to layer $\mathcal{L}_n$, mentioned above, is given by

$$\begin{aligned}\mathcal{C}(\mathcal{L}_n) &= \mathcal{C}(\delta\mathcal{Q}_n[i] \mid \delta\mathcal{Q}_{n-1}[i] \;\&\; \lambda_n(\tilde{x}[i])) \\ &= -\sum_i \log_2 \mathbf{P}[\mathcal{Q}_n(X) - \mathcal{Q}_n(\tilde{X}) = \delta\mathcal{Q}_n[i] \\ &\qquad \mid \mathcal{Q}_{n-1}(X) - \mathcal{Q}_{n-1}(\tilde{X}) = \delta\mathcal{Q}_{n-1}[i] \\ &\qquad \;\&\; \lambda_n(\tilde{X}) = \lambda_n(\tilde{x}[i])] \qquad (13) \\ &= -\sum_i \log_2 \mathbf{P}[\mathcal{Q}_n(X) - \mathcal{Q}_n(\tilde{X}) = \delta\mathcal{Q}_n[i] \\ &\qquad \;\&\; \mathcal{Q}_{n-1}(X) - \mathcal{Q}_{n-1}(\tilde{X}) = \delta\mathcal{Q}_{n-1}[i] \\ &\qquad \mid \lambda_n(\tilde{X}) = \lambda_n(\tilde{x}[i])] \\ &\quad + \sum_i \log_2 \mathbf{P}[\mathcal{Q}_{n-1}(X) - \mathcal{Q}_{n-1}(\tilde{X}) = \delta\mathcal{Q}_{n-1}[i] \\ &\qquad \mid \lambda_n(\tilde{X}) = \lambda_n(\tilde{x}[i])]. \qquad (14)\end{aligned}$$

From (11) we have

$$\mathcal{Q}_{n-1}(x) - \mathcal{Q}_{n-1}(\tilde{x}) = \frac{\mathcal{Q}_n(x) - \mathcal{Q}_n(\tilde{x}) + \lambda_n(\tilde{x})}{K_n} - \frac{\lambda_n(x)}{K_n} \qquad (15)$$

$$= \left\lfloor \frac{\mathcal{Q}_n(x) - \mathcal{Q}_n(\tilde{x}) + \lambda_n(\tilde{x})}{K_n} \right\rfloor, \ \forall x. \qquad (16)$$

Equation (16) follows from (15) because the left hand side is an integer and $\frac{\lambda_n(x)}{K_n} \in [0, 1)$ from (12). According to (16), $\mathcal{Q}_{n-1}(X) - \mathcal{Q}_{n-1}(\tilde{X})$ is completely determined by $\mathcal{Q}_n(X) - \mathcal{Q}_n(\tilde{X})$ and $\lambda_n(\tilde{X})$ and so (14) becomes

$$\mathcal{C}(\mathcal{L}_n) = \mathcal{C}(\delta\mathcal{Q}_n[i] \mid \lambda_n(\tilde{x}[i])) - \mathcal{C}(\delta\mathcal{Q}_{n-1}[i] \mid \lambda_n(\tilde{x}[i])). \qquad (17)$$

It is reasonable to assume that $\delta\mathcal{Q}_n[i]$ is independent of $\lambda_n(\tilde{x}[i])$, however it is not at all reasonable to assume that $\delta\mathcal{Q}_{n-1}[i]$ is independent of $\lambda_n(\tilde{x}[i])$. To see this, consider Fig. 5, for which $K_n = 2$. The assumption of DPCM/CR coding is that small absolute differences are more likely than large absolute differences. It is clear that lower values of $\delta\mathcal{Q}_{n-1}[i]$ are favored when $\lambda_n(\tilde{x}[i]) = 0$ than when $\lambda_n(\tilde{x}[i]) = 1$. Such a relationship does not exist between $\lambda_n(\tilde{x}[i])$ and $\delta\mathcal{Q}_n[i]$. It follows, from (17), that

$$\begin{aligned}\mathcal{C}(\mathcal{L}_n) &= \mathcal{C}(\delta\mathcal{Q}_n[i]) - \mathcal{C}(\delta\mathcal{Q}_{n-1}[i] \mid \lambda_n(\tilde{x}[i])) \\ &> \mathcal{C}(\delta\mathcal{Q}_n[i]) - \mathcal{C}(\delta\mathcal{Q}_{n-1}[i]) \qquad (18)\end{aligned}$$

and so, by induction

$$\sum_{m=1}^{n} \mathcal{C}(\mathcal{L}_m) > \mathcal{C}(\delta\mathcal{Q}_n[i]). \qquad (19)$$

Equation (19) states that our efficiency goal cannot be achieved for layered DPCM/CR coding. The number of bits required to encode the first n quantization layers in the layered DPCM/CR scheme exceeds the number of bits required to encode the DPCM/CR symbols associated with the quantizer $\mathcal{Q}_n$ alone. This unfortunate result arises from the order of transmission. The inequality of (19) could be made an equality only if the value of $\lambda_n(\tilde{x}[i])$ could be employed in coding $\delta\mathcal{Q}_{n-1}[i]$. This would require the first n quantization layers to be decoded for the reference sample, $\tilde{x}[i]$, before the first $n-1$ layers could be decoded for the current sample, $x[i]$. Such a decoding order, however, is not compatible with our layered coding philosophy, in which each quantization layer should be completely independent of all subsequent layers.

Regarding the probability table sizes for the conditional arithmetic coding represented by (13), similar considerations apply to those discussed in Section III-A, except for the additional conditioning on $\lambda_n(\tilde{x}[i])$. From (12), $\lambda_n(\tilde{x}[i])$ can take on any of K_n possible values, increasing the number of probability table entries accordingly. Consequently our layered DPCM/CR probability tables require a total of no more than $2K_n$ times as many entries as the probability table required for a conventional DPCM/CR system, encoding the single sequence of DPCM/CR symbols, $\delta\mathcal{Q}_N[i]$. In practice we select $K_n = 2$ so that *each successive quantization layer doubles the precision of the subband sample values.*

2) Behavior of Layered DPCM/CR with Exponential Model Statistics: Despite the fact that our layered DPCM/CR scheme cannot achieve our coding efficiency goal, it nevertheless performs considerably better than simple bit-planing. In this section we confirm this claim in the case of a source whose nth layer difference sequence, $\delta\mathcal{Q}_n[i]$, is modeled by an exponential distribution

$$\mathbf{P}\Big(\mathcal{Q}_n(X) - \mathcal{Q}_n(\tilde{X}) = j\Big) = \frac{1 - r^{-1}}{1 + r^{-1}} \cdot r^{-|j|}. \qquad (20)$$

Our experimental observations suggest that the exponential distribution is a very good model for DPCM sequences, $\delta\mathcal{Q}_n[i]$, obtained by uniformly quantizing the low frequency spatiotemporal subband. By making the reasonable assumption that $\mathbf{P}(\lambda_n(\tilde{X}) = 0) = \mathbf{P}(\lambda_n(\tilde{X}) = 1) = \frac{1}{2}$, it may be shown that the average number of bits per symbol, required to encode $\delta\mathcal{Q}_n[i]$, $\delta\mathcal{Q}_{n-1}[i]$ and $\mathcal{L}_n$ are given by the entropies

$$\begin{aligned}\mathcal{H}(\delta\mathcal{Q}_n) &= \frac{2r^{-1}}{1 - r^{-2}} \log_2 r + \log_2\left(\frac{1 + r^{-1}}{1 - r^{-1}}\right) \qquad (21) \\ \mathcal{H}(\delta\mathcal{Q}_{n-1}) &= \frac{2r^{-1}}{1 - r^{-2}} \log_2 r + r^{-1} \log_2\left(\frac{2r^{-1}}{1 - r^{-1}}\right)\end{aligned}$$

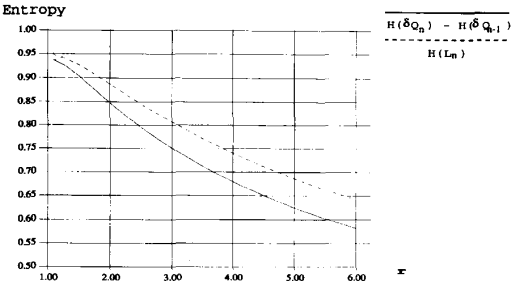

Fig. 6. Entropy of quantization layer $\mathcal{L}_n$ compared with difference between entropies of $\delta\mathcal{Q}_n$ and $\delta\mathcal{Q}_{n-1}$, for the exponential source model of (20).

$$+ \log_2\left(\frac{1}{1-r^{-1}}\right) \tag{22}$$

$$\mathcal{H}(\mathcal{L}_n) = \frac{1}{1+r}\log_2(1+r) + \frac{1}{1+r^{-1}}\log_2(1+r^{-1}). \tag{23}$$

Equations (21)–(23) are used to generate the curves of Fig. 6 for various values of the model parameter r. These curves compare the difference between the entropies of $\delta\mathcal{Q}_n$ and $\delta\mathcal{Q}_{n-1}$ with $\mathcal{H}(\mathcal{L}_n)$, the average number of bits actually required to code each symbol of $\mathcal{L}_n$ in our layered DPCM/CR system. The figure bears out the conclusion of (18), while at the same time demonstrating that performance can considerably exceed that of simple bit-planing, for which $\mathcal{H}(\mathcal{L}_n) = 1$ for all r.

C. Layered Zero Coding

By far the most frequent symbol coded by the PCM and DPCM/CR strategies discussed in Sections III-A and III-B is the zero symbol. In the PCM case the zero symbol is generated when $\mathcal{Q}_n(x[i]) = 0$ and corresponds to *inactivity* in the high frequency subband being coded. In the DPCM/CR case the zero symbol is generated when the difference signal $\delta\mathcal{Q}_n[i] = 0$, corresponding to *smooth* regions in the low frequency subband being coded. The assumption, stated in Sections III-A and III-B, of statistical independence between the elements of the sequences $\mathcal{Q}_n(x[i])$ and $\delta\mathcal{Q}_n[i]$, respectively, is unreasonable in these *inactive* or *smooth* regions, whose pixels typically exhibit strong spatial correlation. For this reason, it is usual to separate the coding of subband sample values into two phases. The first phase codes whether the sample is zero or nonzero, relative to quantizer $\mathcal{Q}_n$— i.e., whether $\mathcal{Q}_n(x[i])$ (resp. $\delta\mathcal{Q}_n[i]$) is zero or nonzero. The second phase then codes the particular value of each nonzero symbol, using the techniques developed in Sections III-A and III-B. The first phase, which we shall refer to as zero coding, attempts to exploit spatial, or some other dependencies amongst the subband samples. The most commonly employed technique is run-length coding, a strategy used for the encoding of facsimile messages.

In Section III-C-1 we present a more efficient zero coding strategy than run-length coding, which is also well-suited to our layered quantization approach. Then, in Section III-C-2 we extend this zero coding technique to exploit statistical dependencies between subbands.

1) Zero Coding to Exploit Full Spatial Correlation: The object of zero coding is to efficiently encode the sequence, $\sigma_n[i]$, given by

$$\sigma_n[i] = \begin{cases} 1, \text{ if } \mathcal{Q}_n(x[i]) \neq 0, & \text{else } 0 \quad \text{PCM} \\ 1, \text{ if } \delta\mathcal{Q}_n[i] \neq 0, & \text{else } 0 \quad \text{DPCM/CR} \end{cases}. \tag{24}$$

In run-length coding, this sequence is coded implicitly via so-called white and black run lengths, representing the lengths of contiguous ones and contiguous zeros, respectively. As such, run-length coding is only able to take advantage of spatial correlation in one dimension. $\sigma_n[i]$, however, is a 1-D scanning of a multidimensional signal. In order to take advantage of 2-D dependencies amongst the elements of $\sigma_n[i]$, we employ a sequence of "conditioning terms," denoted by $\kappa_n[i]$. The idea is to use conditional arithmetic coding, to encode the sequence $\sigma_n[i]$, conditioned upon these $\kappa_n[i]$ terms. In general, each term, $\kappa_n[i]$, is a function of the quantized values of subband samples in the neighborhood of $x[i]$. Naturally, $\kappa_n[i]$ may be based only on information available to the decoder prior to decoding $\sigma_n[i]$. Our proposed definition of $\kappa_n[i]$ is provided presently, however we first indicate how these conditioning terms are employed in a layered coding system.

Given the quantizer constraints represented by (6), (7), and (10), zero sequences associated with successive quantization layers are strongly correlated by the relationship

$$\sigma_n[i] \geq \sigma_{n-1}[i]$$

This means that once a sample has been shown to be nonzero in quantization layer $\mathcal{L}_{n-1}$ there is no need to code whether or not it is zero in quantization layer $\mathcal{L}_n$. Thus, when layered zero coding is used with layered PCM coding, for example, the total number of bits required to code layer $\mathcal{L}_n$ is

$$\begin{aligned} C(\mathcal{L}_n) &= -\sum_{i \ni \sigma_{n-1}[i]=0} \log_2 \mathbf{P}(\sigma_n[i] \mid \sigma_{n-1}[i] = 0 \;\&\; \kappa_n[i]) \\ &\quad - \sum_{i \ni \sigma_n[i]=1} \log_2 \mathbf{P}(\mathcal{Q}_n(x[i]) \mid \sigma_n[i] = 1 \;\&\; \mathcal{Q}_{n-1}(x[i])). \end{aligned} \tag{25}$$

The first term in (25) represents the number of bits required to code the zero sequence, $\sigma_n[i]$, whereas the second term represents the number of bits required to code the refinement of the nonzero values to quantization precision, $\mathcal{Q}_n$. In Section III-A we proved that no loss in compression efficiency is suffered by using layered PCM coding instead of conventional single quantizer PCM. It would be nice to be able to prove a similar result when PCM coding is combined with zero coding, as given by (25). In the degenerate case of $\kappa_n[i]$ a constant, the result may readily be obtained by combining (5), (24), and (25). In the more useful case, however, when $\kappa_n[i]$ is anything but constant, we are not able to prove anything conclusive about the compression performance of the overall layered system. In practice, though, we find that the layered system almost always exhibits superior compression performance to an equivalent single quantizer system. The

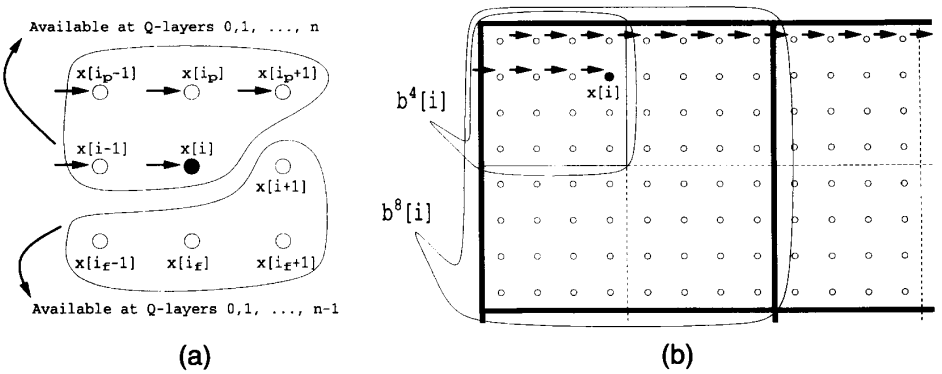

Fig. 7. Pixels used to build the spatial conditioning term, $\kappa_n[i]$, used in coding whether sample $\mathcal{Q}_n(x[i])$ is zero or not: (a) local scale; (b) wide scale. Arrows indicate the lexicographic scanning order, followed in coding the zero sequence.

reason for this is that the layered system is able to exploit spatial correlation at a variety of quantization precisions via layered zero coding, whereas a single layer system is able to exploit spatial correlation only in coding the zero sequence associated with a single quantizer. We find, therefore, that in practice we almost always exceed our multilayer efficiency goal. Results presented in Section IV-B demonstrate that this is true even for the DPCM/CR subband, for which, in the absence of zero coding, the layered system is guaranteed to have poorer compression performance than a single quantizer system—see Section III-B.

We turn our attention now to the definition of the conditioning terms, $\kappa_n[i]$. Although numerous possibilities present themselves, the definition proposed here has been found to offer good compression performance with a relatively simple implementation. For PCM quantization and coding we adopt a fixed lexicographic scanning order for the sequence of sample values, $x[i]$, and hence $\sigma_n[i]$. We describe the generation of the $\kappa_n[i]$ terms only in the PCM case. A straightforward modification of this description is required for the DPCM/CR case, in which the zig-zag scanning order mentioned in Section III-B is applied.

In our proposed definition for $\kappa_n[i]$, the basic idea is to first consider the values of samples adjacent to $x[i]$. We refer to this as local scale conditioning. Only if all of these samples are found to be zero do we resort to wider scale conditioning. Fig. 7(a) and 7(b) provide frameworks for understanding the local and wide scale approaches, respectively. We first collect local conditioning information in $\kappa'_n[i]$, defined by

$$\begin{aligned}\kappa'_n[i] = {} & \sigma_n[i-1] + 2\sigma_n[i_p] + 4(\sigma_n[i_p - 1] \,\imath\, \sigma_n[i_p + 1]) \\ & + 8 \cdot \sigma_{n-1}[i+1] + 16 \cdot \sigma_{n-1}[i_f] \\ & + 32(\sigma_{n-1}[i_f - 1] \,\imath\, \sigma_{n-1}[i_f + 1]) \end{aligned} \tag{26}$$

where $\imath$ denotes logical "or", and i_p and i_f denote the indices of adjacent samples to $x[i]$ on the previous and future scan lines respectively, as depicted in Fig. 7(a). The binary representation of $\kappa'_n[i]$, as defined in (26), consists of 6 binary digits. Each of $x[i]$'s most immediate neighbors, $x[i-1], x[i+1], x[i_p]$ and $x[i_f]$, corresponding to the first, fourth, second and fifth terms in the right hand side of (26), respectively, independently determines the value of a single digit in the binary representation of $\kappa'_n[i]$. The corner neighbors, $x[i_p-1], x[i_p+1], x[i_f-1]$ and $x[i_f+1]$, however, are grouped into pairs, corresponding to the third and sixth terms in (26), via the logical "or" operator. These pairs determine the remaining two binary digits of $\kappa'_n[i]$. This pairing is to limit the total amount of conditioning information and hence the size of the conditional probability tables in a practical implementation.

When $\kappa'_n[i] = 0$, we resort to wider scale conditioning information, which is based on fixed blocks of 4×4 and 8×8 samples. In particular, let $b^4[i]$ be the collection of sample indices, j, belonging to the 4×4 block containing the sample $x[i]$, as depicted in Fig. 7(b). Similarly, let $b^8[i]$ be the collection of sample indices, j, belonging to the 8×8 block containing the sample $x[i]$. We define wide scale conditioning terms, $\mathcal{B}^4_n[i]$ and $\mathcal{B}^8_n[i]$ by

$$\mathcal{B}^4_n[i] = \max_{i>j\in b^4[i]} \sigma_n[j] \,\imath\, \max_{i<j\in b^4[i]} \sigma_{n-1}[j]$$

and

$$\mathcal{B}^8_n[i] = \max_{i>j\in b^8[i]} \sigma_n[j] \,\imath\, \max_{i<j\in b^8[i]} \sigma_{n-1}[j].$$

The local and wide scale terms are combined to form our ultimate sequence of conditioning terms, $\kappa_n[i]$, according to

$$\kappa_n[i] = \begin{cases} 2 + \kappa'_n[i] & \text{if } \kappa'_n[i] \neq 0 \\ \max\left\{2\mathcal{B}^4_n[i], \mathcal{B}^8_n[i]\right\} & \text{if } \kappa'_n[i] = 0 \end{cases}. \tag{27}$$

In practice, we find that most of the coding gain is obtained from conditional coding of $\sigma_n[i]$ relative to the local scale term, $\kappa'_n[i]$. The incorporation of wide scale terms, $\mathcal{B}^4_n[i]$ and $\mathcal{B}^8_n[i]$, in (27) typically reduces the bit rate by only a few percent, however an efficient implementation of the wide scale conditioning has negligible impact on the computational demands of our algorithm, justifying its incorporation.

2) Exploiting Correlation between Subbands: In this section we propose an extension of the conditioning terms, $\kappa_n[i]$, presented in section III-C-1, so as to exploit correlation existing between PCM coded subbands. Shapiro [12] has proposed a 2-D subband coding algorithm for images, which

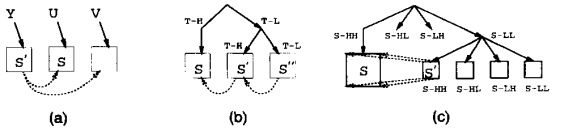

Fig. 8. Intersubband correlation relationships. (a) Between color components. (b) Between subbands of the same temporal hierarchy. (c) Between subbands of the same spatial hierarchy. $\mathcal{S}'$ denotes the reference subband used in coding subband $\mathcal{S}$. T—L, T—H, S—LL, S—LH, etc., have the same interpretation as in Figs. 2 and 3.

takes advantage of correlation between subbands at different levels in a spatial decomposition hierarchy. In our proposed 3-D structure, however, there also exists the possibility of exploiting correlation between different temporal subbands. In addition, a color system has the opportunity to exploit correlation between luminance and chrominance subbands with identical spatiotemporal passbands. Each of these three types of intersubband relationship are illustrated in Fig. 8(a)–8(c).

For each subband, $\mathcal{S}$, we attempt to identify a reference subband, $\mathcal{S}'$, which may be used to provide additional conditioning information for coding the zero sequences associated with $\mathcal{S}$. More than one of the potential reference subband relationships indicated in Figs. 8(a)–8(c) will often be available. In such cases we select $\mathcal{S}'$ to be a luminance subband with identical spatiotemporal passband to that of $\mathcal{S}$, if possible, as in Fig. 8(a). Naturally, this can only be possible if $\mathcal{S}$ is a chrominance subband. Otherwise we select $\mathcal{S}'$ to be a subband with the same spatial passband and color component type as $\mathcal{S}$ but one level down in the temporal hierarchy, if possible, as in Fig. 8(b). Failing these first two possibilities, we select $\mathcal{S}'$ to be a subband with identical temporal passband and color component type to that of $\mathcal{S}$, but one level down in the spatial subband hierarchy, as in Fig. 8(c). This prioritization of the three potential intersubband relationships depicted in Figs. 8(a)–8(c) is based on empirical observations of the correlation between various subbands. In some instances, no reference subband, $\mathcal{S}'$, is available, in which case the definition of $\kappa_n[i]$ is unaltered from that of Section III-C-1. This procedure for assigning reference subbands is best illustrated by the subband structure diagrams in Figs. 2 and 3, wherein the relationships are indicated by dashed arrows from reference to referee subbands.

In order to take advantage of correlation with subband $\mathcal{S}'$ in coding the zero sequence, $\sigma_n[i]$, of subband $\mathcal{S}$, the first step is to obtain a reference zero sequence, $\sigma'_n[i]$, with the same number of elements as $\sigma_n[i]$, and based on quantized samples of $\mathcal{S}'$. The assumption is that $\sigma_n[i]$ and $\sigma'_n[i]$ are highly correlated. In the absence of camera pan motion, for the correlation relationships of Figs. 8(a) and 8(b), each sample, $x[i]$, of subband $\mathcal{S}$ should be best correlated with the sample $x'[i]$ of subband $\mathcal{S}'$, i.e., the sample with identical index in the lexicographic scanning sequence. In this case we set

$$\sigma'_n[i] = 1, \text{ if } \mathcal{Q}'_{n'}(x'[i]) \neq 0, \text{ else } 0$$

where $\mathcal{Q}'_{n'}$ is the quantizer for layer $\mathcal{L}_{n'}$ of subband $\mathcal{S}'$, with $n' \leq n$ selected so that the quantization step size of $\mathcal{Q}'_{n'}$ is as close as possible to that of $\mathcal{Q}_n$.[7] If the intersubband relationship is of the type shown in Fig. 8(c), the samples of $\mathcal{S}'$ must first be duplicated horizontally and vertically because $\mathcal{S}'$ has only one quarter as many samples as $\mathcal{S}$. When camera pan motion is involved, reference subband samples must first be appropriately shifted to compensate. The appropriate shifts are readily computed from the camera pan parameters. Samples in $\mathcal{S}$, for which no suitable reference sample in $\mathcal{S}'$ exists, after the application of such shifts, may only exploit the spatial correlation described in Section III-C-1.

We are now in a position to augment the definition of $\kappa_n[i]$ given in Section III-C-1 to include conditioning on the reference zero sequence $\sigma'_n[i]$. In particular, we define

$$\kappa_n[i] = \max\{m \mid 0 \leq m \leq \min\{n, 3\} \ \& \ \sigma'_{n-m}[i] = 1\} + 4\kappa_n^s[i]$$

where $\kappa_n^s[i]$ is the value of $\kappa_n[i]$ given in (27), for purely spatial conditioning. In this way the conditional probability tables required for full spatial and intersubband conditioning in coding $\sigma_n[i]$ must contain $4 \times 66 = 246$ entries.

Before concluding this section we note that, when intersubband conditioning is used, layer $\mathcal{L}_n$ of subband $\mathcal{S}$ may be successfully decoded only if quantization layer $\mathcal{L}_{n'}$ of subband $\mathcal{S}'$ is available. The condition $n' \leq n$ stated above allows us to guarantee that all available quantization layers of subband $\mathcal{S}$ may be successfully decoded, provided at least as many quantization layers of subband $\mathcal{S}'$ are available. Our proposed rate limiting strategy, presented in Section IV-A enforces this constraint.

IV. Experimental Investigation

The purpose of this section is to present results indicating the performance of our multirate subband coding algorithm. In Section IV-A we present some details concerning our specific implementation of the algorithm, in order to provide a context for the results which appear in Section IV-B.

A. Implementation Details

Sections II and III reveal that our embedded multirate, multiresolution bit stream is composed of a large number of subbands, each of which is to be transmitted in a sequence of quantization layers. Two important issues, however, remain to be clarified: the organization of these quantization layers and subbands within a realistic transmission sequence; and the selection of appropriate subsets of the bit stream to accommodate a variety of decoder display format and bandwidth constraints. Important as they are, we regard these as implementation issues and now briefly discuss how they are resolved in our experimental context.

The actual bit stream produced by our encoder implementation is described in terms of two basic constructs, which we refer to as the *code block* and the *transmission block*.

[7] The context for this observation is that all the subband filter impulse responses are normalized so as to have dc gain of $1/\sqrt{2}$, i.e., approximately unit power gain. Filter normalization affects the relative magnitudes of subband samples in $\mathcal{S}'$ and $\mathcal{S}$ and hence the relationship between $\mathcal{Q}'_{n'}$ and $\mathcal{Q}_n$.

Each code block contains a single, independent arithmetic code word, which encodes all N quantization layers, for a fixed number of samples of a particular subband. Code blocks are then collected into transmission blocks, each of which represents a fixed number of frames, F, of the video sequence. For the results presented in Section IV-B, $F = 32$.[8]

Specifically, each code block encodes 2^i frames of subband samples, where i is selected to be as large as possible so that the number of subband samples encoded by the code block does not exceed some constant, P, to be discussed shortly, and 2^i does not exceed the number of transmission block frames, F. Subbands at the top of the spatial decomposition hierarchy contain many samples per frame, in which case i may be negative, indicating that several code blocks are used to encode each subband frame. Other code blocks may represent many frames worth of subband samples. The arithmetic message contains an encoding of quantization layer $\mathcal{L}_1$ for all samples, followed by $\mathcal{L}_2$ for all samples and so on. Each code block also contains a header specifying the number of bytes of the arithmetic message which must be received in order to unambiguously decode the first n quantization layers, for $n = 1, 2, \cdots, N$. The constant, P, governing code block size, should be large enough to prevent this header information from having a significant impact on the overall bit rate. Apart from this consideration, however, smaller values of P are to be preferred, as they increase the number of code blocks, which in turn increases opportunity for effective bit rate control and for large scale parallelism in a real time implementation. Our experiments are conducted with a value of $P = 8000$ samples.

The transmission block is the basic unit of synchronization and rate control for our bit stream. In addition to the code blocks themselves, each transmission block also contains camera pan vectors and information identifying the subband structure used for encoding. The multiresolution property of the bit stream arises from the fact that only the code blocks which represent subbands of interest to the decoder need be received and decoded. The multirate property arises from the fact that the number of quantization layers contained in each code block may be independently decreased at any point in the distribution of the bit stream, subject to the intersubband constraints discussed in Section III-C-2.

We implement a very simple algorithm for bit rate control. We first translate a limit, $\mathcal{B}_r$, on the bit rate to a limit, $\mathcal{T}_B = (F \cdot \mathcal{B}_r)/(8 \cdot \text{frame rate})$, on the number of bytes for each transmission block. For each transmission block we then examine the headers associated with every code block in the transmission block[9] to determine the number n_l such that n_l quantization layers may be retained for each code[10] block without exceeding the limit, $\mathcal{T}_B$. Finally, after n_l layers of all code blocks have been incorporated, code blocks are considered one at a time as candidates for an extra quantization layer, so as to utilize as much as possible of the $\mathcal{T}_B$ bytes available for the transmission block. In this process, code blocks are considered in order of increasing spatial subband bandwidth and, subordinately, in order of increasing temporal subband bandwidth. Amongst candidates of equal spatial and temporal subband bandwidth, code blocks are considered in decreasing order of the number of bytes required for quantization layer $\mathcal{L}_{n_l+1}$, the assumption being that those code blocks which require more bytes to encode layer $\mathcal{L}_{n_l+1}$ contain more important visual information. In addition, we do not permit a code block to be allocated $n_l + 1$ quantization layers when a code block on which it depends for intersubband correlation information, as discussed in Section III-C-2, has only been allocated n_l quantization layers.

For spatial subband filters we select the symmetric, nearly orthogonal, nearly perfect reconstruction, nine tap filters of Adelson *et al.* [1]. The filter impulse responses are normalized to have dc gain of $\frac{1}{\sqrt{2}}$, corresponding to a unit 2-norm. With this normalization, the work of Pearlman [9] predicts that optimal performance, in the mean squared error sense, should be obtained by using the same set of N quantizers for each subband. Our experimental results in Section IV-B are based on this selection. The same set of $N = 9$ quantizers, having step sizes of $\Delta_1 = 256, \Delta_2 = 128, \cdots, \Delta_9 = 1$, is assigned to each luminance subband. Each chrominance subband is also assigned an identical set of $N = 9$ quantizers, this time with step sizes of $\Delta_1 = 200, \Delta_2 = 100, \cdots, \Delta_9 = 0.78125$.

TABLE III
PERFORMANCE OF OUR ALGORITHM RESTRICTED TO TWO DIMENSIONS, FOR MULTIRATE, MULTIRESOLUTION STILL IMAGE COMPRESSION OF 512 × 512 GREY SCALE 'LENA' TEST IMAGE; COMPARISON WITH HIGH-PERFORMANCE MULTIRATE, FIXED-RESOLUTION IMAGE COMPRESSION ALGORITHMS BY SHAPIRO [12] AND SAID AND PEARLMAN [11]; CPU DECODING TIMES REPORTED FOR UNI-PROCESSOR SPARC 10/41

	Shapiro	Said and Pearlman		Taubman and Zakhor	
Bit Rate (bits/pixel)	PSNR (dB)	PSNR (dB)	CPU* (s)	PSNR (dB)	CPU† (s)
1.0	39.55	40.04	4.2	40.35	3.0
0.5	36.28	36.84	2.0	37.25	2.6
0.25	33.17	33.67	0.9	34.12	2.2
0.125	30.23	–	–	30.96	1.7

*Does not include subband transform; †All-inclusive

B. Results

We begin this section by briefly presenting the performance of our multirate compression algorithm when restricted to two dimensions. This is of interest because it demonstrates a more general multimedia applicability of our algorithm. We apply our algorithm to the compression of a 512 × 512 grey scale version of the well-known test image, 'Lena', using the luminance component of the subband structure of Fig. 2, and building probability tables for arithmetic coding from the ensemble statistics of 30 natural images, not including 'Lena'. Applying the rate limiting algorithm discussed in

[8] Naturally, we must have $F = 1$ for still image compression.

[9] In practice these headers are all sent at the beginning of the transmission block for convenience.

[10] Concerns over the wisdom of transmitting the same number of quantization layers for all code blocks should be allayed by the observation that the first few quantization layers for subbands with small signal energy may easily consist of no nonzero samples and hence require no bits to encode. This behavior is determined by the quantization step sizes selected for each subband.

TABLE IV
COMPARISON OF MULTIRATE SUBBAND COMPRESSION WITH MPEG-1; 'MPEG' COLUMNS INDICATE THE OVERALL PSNR VALUES OBTAINED USING MPEG; 'NO PAN' COLUMNS INDICATE THE IMPROVEMENT IN OVERALL PSNR VALUES OVER MPEG, OBTAINED USING OUR SUBBAND SYSTEM *without* CAMERA PAN COMPENSATION; 'PAN' COLUMNS INDICATE THE IMPROVEMENT IN OVERALL PSNR VALUES OVER MPEG, FOR THE SUBBAND SYSTEM *with* CAMERA PAN COMPENSATION

		500 kb/s			1.0Mb/s			1.5 Mb/s		
Video Sequence	Component	MPEG (db)	No Pan (dB)	Pan (dB)	MPEG (db)	No Pan (dB)	Pan (dB)	MPEG (db)	No Pan (dB)	Pan (dB)
'football'	Y	28.62	+0.38	+0.94	31.46	+0.10	+0.96	33.26	+0.19	+1.37
138 frames	U	36.22	+0.96	+1.08	38.28	+0.54	+0.75	39.34	+0.39	+0.87
panning	V	33.34	+0.97	+1.18	35.92	+0.75	+1.04	37.32	+0.52	+1.04
'pingpong'	Y	28.39	−1.94	+1.23	31.15	−1.43	+1.30	32.80	−0.52	+1.82
264 frames	U	36.11	−1.57	−0.59	38.19	−1.59	−0.07	39.34	−0.98	+0.46
zoom, pan, still	V	37.20	−1.00	+0.79	39.16	−0.97	+0.60	40.17	−0.37	+0.92
'flower garden'	Y	22.02	−2.76	−1.54	25.43	−3.72	−2.15	27.16	−3.98	−2.39
120 frames	U	30.95	−0.79	−0.16	32.92	−1.94	−1.07	34.12	−2.55	−1.32
translating	V	27.87	−1.61	−0.84	30.58	−3.18	−1.63	32.25	−3.66	−2.51

Section IV-A to a single encoded bit stream, the various bit rates and corresponding reconstruction peak signal to noise ratios (PSNR)[11] appearing in Table III are obtained. The table compares our results against those obtained by Shapiro [12] and Said and Pearlman [11], based around a compression strategy originally developed by Shapiro which produces a bit stream with the multirate property, but not the multiresolution property. It is noteworthy that our algorithm exhibits improved compression performance as well as reduced computational requirements, over the algorithms of [12] and [11].

In Table IV, we present a comparison of our multirate video compression algorithm with an MPEG-1 implementation obtained from Bellcore, using three standard video test sequences, 'pingpong,' 'football,' and 'flower garden' at SIF525 resolution[12] and three different bit rates. The MPEG encoder builds each group of pictures (GOP) from one intra coded frame, four predicted frames and ten bidirectionally predicted frames, using half pixel motion vector resolution over the CCITT recommended search ranges [3] of ±7 pixels per frame for 'pingpong' and 'flower garden' and ±15 pixels per frame for 'football'. The subband results are obtained using the structure of Fig. 3 and an eighth pixel resolution camera pan search range of ±30 pixels/frame.[13] Due to a lack of suitable video source material for building independent probability tables for arithmetic coding, all three sequences were used to build ensemble statistics which were then used to encode the same three sequences. It is important to realize that a separate bit stream must be generated by the MPEG-1 encoder for each of the three bit rate conditions presented in Table IV, whereas the subband encoder generates just a single bit stream, from which the rate-limiting strategy of Section IV-A is used to select appropriate subsets for decoding. This rate limiting strategy is so effective that, for bit rates in excess of 30 kb/s, the number of bytes, $\mathcal{T}_A$, actually allocated to each transmission block is always found to lie in the range $\mathcal{T}_B \geq \mathcal{T}_A > 0.999\mathcal{T}_B$, where $\mathcal{T}_B$ is determined from the bit rate limit as described in Section IV-A.

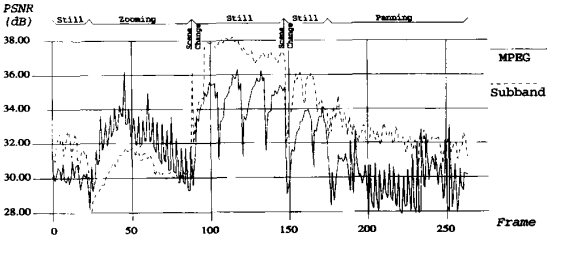

Fig. 9. Frame-by-frame distortion for luminance component of the 'pingpong' sequence, reconstructed from 1.0 Mb/s MPEG bit stream and from the subband bit stream, rate limited to 1.0 Mb/s. Annotations indicate camera motion present during each segment of the sequence.

Further insight into the comparative performance of our multirate algorithm may be obtained by considering the frame-by-frame reconstruction error for the 'pingpong' sequence at 1 Mb/s, as presented in Fig. 9. Clearly, the subband system significantly outperforms MPEG during regions when the camera is still or else panning. During camera zoom, however, the motion compensation approach of MPEG is definitely superior. A similar effect is observed with the 'flower garden' sequence, in which camera translation leaves every part of the scene in motion, which cannot be described by our simple camera pan model. It is interesting, however, that our algorithm outperforms MPEG in compressing the 'football' sequence which contains a high degree of complex foreground motion. It can be argued that panning is by far the most common form of camera motion experienced in entertainment video. The comparison of the performance of our algorithm with and without camera pan compensation, also appearing in Table IV, clearly demonstrates the effectiveness of this feature.

In order to put the multirate aspect of our compression algorithm into perspective, Fig. 10 illustrates the mean luminance and chrominance distortion, expressed in PSNR, as a function of maximum allowable bit rate, for the 'pingpong' sequence.

[11] $\text{PSNR} \triangleq 10 \log_{10} \frac{255^2}{\text{Mean Squared Error}}$

[12] 30 progressively scanned 352 × 240 frames per second, obtained by horizontally filtering and downsampling odd fields, starting with 4:2:2-525 interlaced format. Chrominance components are additionally subsampled by two both horizontally and vertically

[13] Note that camera pan parameter estimation time is independent of search window size, due to the FFT approach discussed in Section II-B.

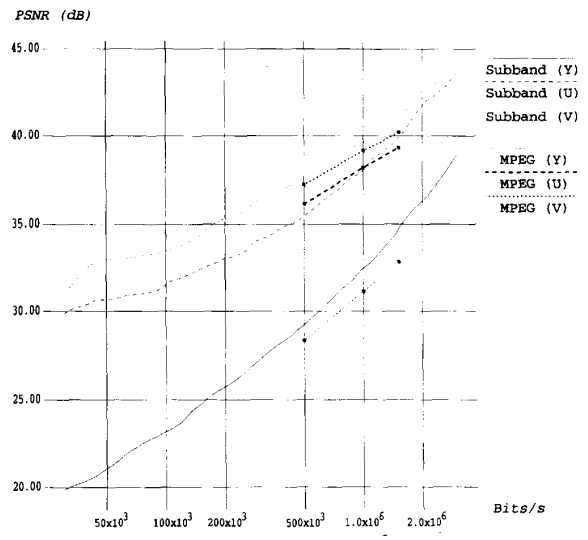

Fig. 10. Rate-distortion curves for 'pingpong' sequence. Overall PSNR values for Y, U, and V components are plotted against the bit rate limit imposed upon the multirate bit stream prior to decoding. MPEG distortion values from Table IV are also plotted for reference.

Fig. 11. Frame 210 from 'pingpong' sequence, decoded at 1.5Mb/s from multirate subband bit stream.

Measured points on the MPEG rate-distortion curve are also shown. Fig. 10 demonstrates the capacity of our multirate algorithm to smoothly exchange bit rate for reconstruction quality. Figs. 11–13 reveal the effect of such rate/quality trade-offs on the visual appearance of the reconstructed video sequence. The figures display frame 210 from the panning segment of the 'pingpong' sequence, decoded at bit rates of 60 kb/s, 300 kb/s and 1.5 Mb/s. Fig. 14, displaying the same frame, reconstructed from a 300 kb/s MPEG bit stream, may be compared with Fig. 12 for a visual demonstration of the relative compression performance of our subband algorithm.

Fig. 15 reveals the remarkable fact, alluded to in Section III-C-1, that it is possible to encode the first n quantization layers of a subband, corresponding to quantizers, $\mathcal{Q}_1, \cdots, \mathcal{Q}_n$, with less bits than are required to encode the same subband

Fig. 12. Frame 210 from 'pingpong' sequence, decoded at 300kb/s from multirate subband bit stream.

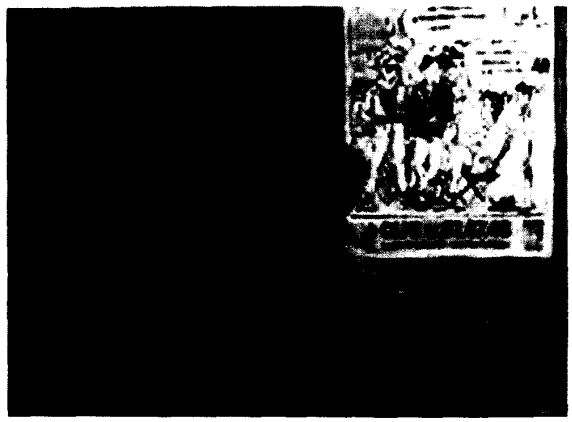

Fig. 13. Frame 210 from 'pingpong' sequence, decoded at 60kb/s from multirate subband bit stream.

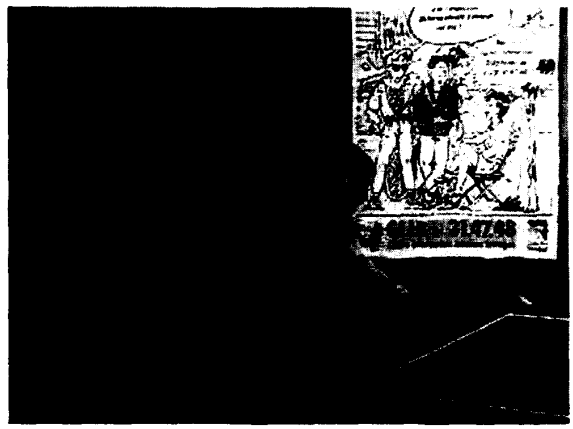

Fig. 14. Frame 210 from 'pingpong' sequence, decoded at 300kb/s from fixed-rate MPEG bit stream.

samples using the single quantizer, $\mathcal{Q}_n$. The figure indicates the reduction in average bit rate obtained by the layered coding strategy versus the average bit rate obtained by using only

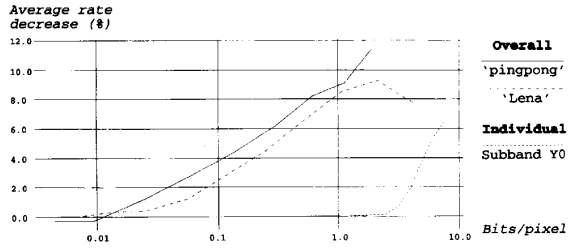

Fig. 15. Decrease in average bit rate achieved by the multirate subband system instead of using conventional single layer quantization. Improvements in overall bit rate are indicated for the'pingpong' sequence and the still image 'Lena'. Improvements in the average bit rate associated with the DPCM/CR coded subband, Y0, are also indicated, for the 'pingpong' sequence.

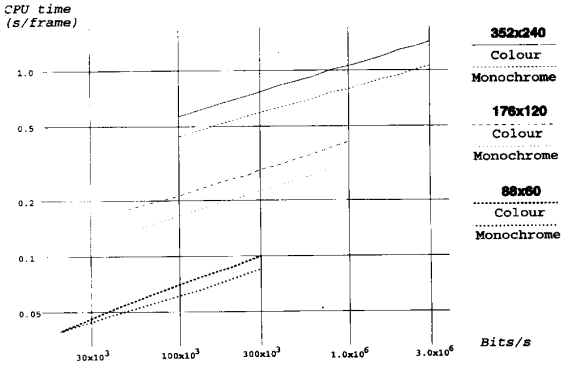

Fig. 16. Decoding CPU time per frame, measured on uniprocessor Sparc 10/41 for various bit rates and display resolutions.

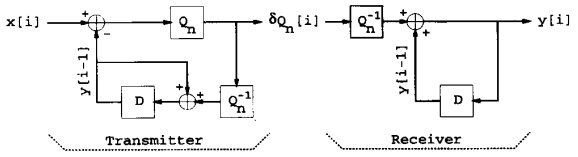

Fig. 17. DPCM transmitter and receiver. D denotes single sample delay. Q_n denotes quantization. Q_n^{-1} denotes inverse quantization.

a single quantization layer, for fixed values of n from 1 to N, where $N = 9$ in our case. Improvements are indicated for the overall bit rates observed during encoding of the 'pingpong' sequence and the still image, 'Lena'. In Section III-B we proved that layered DPCM/CR coding alone is unable to compete with single-layer DPCM/CR coding. Nevertheless, when coupled with layered zero coding, the DPCM/CR coded subband, $Y0$, demonstrates the reverse trend, as indicated in Fig. 15, for the 'pingpong' sequence.

Finally, Fig. 16 provides an indication of computational demands for the algorithm, as a function of bit rate and display resolution, based on our implementation on a uniprocessor Sparc station 10/41. The plotted decoding CPU times do not take into account the need to invert any fractional pixel frame shifts introduced during camera pan compensation—see Section II-B. On the Sparc 10/41 CPU, this operation consumes an average of 2.2 s for each 352 × 240 color frame during panning, regardless of bit rate. The inverse shift, however, may be completely omitted at the expense of a slight, apparent camera jitter (within ±0.5 pixels) during camera pan. Moreover, if the jitter is unacceptable, an approximate inverse shift, with lower computational demands, may well suffice, as presented in [13]. It is for this reason that the time required to invert pan compensating shifts is not included in Fig. 16. The figure reveals a third aspect of scalability to our compression algorithm, namely complexity scalability, whereby reconstruction quality may be traded for decoder complexity by adjusting either the decoded bit rate or the decoded resolution, or a combination of both. It is important to note that the ability to separately encode and decode each of the code blocks discussed in Section IV-A introduces the possibility of a highly parallel hardware or software implementation.

V. Conclusion

We have presented a video coding algorithm, based on 3-D subband decomposition with camera pan compensation and progressive coding of each subband into a set of quantization layers. The encoded bit stream possesses both multiresolution and multirate properties. That is to say that subsets of the bit stream may readily be extracted at any point in its distribution in order to satisfy prevailing constraints on transmission bandwidth and/or desired display format. When camera motion conforms to a pan model, as is common, the algorithm has been shown to outperform conventional techniques in compression efficiency. We stress, however, that its primary value lies not so much in compression efficiency as it does in rate scalability. The multirate property makes our proposed algorithm suitable for a wide variety of applications, for which it is desirable that video encoding proceed independently of distribution bandwidth constraints. In particular, although our results presented in Section IV-B focus on fixed rate subsets of the multirate bit stream, future research will be devoted to variable rate distribution, for which the multirate property is essential to the introduction of reliable statistical multiplexing techniques. Additionally, the proposed algorithm provides excellent still image compression performance for multimedia applications, and is highly amenable to parallel hardware or software implementation.

Appendix
Unwrapping Feedback Loop for Uniformly Quantized DPCM

We show that in the special case of uniformly quantized DPCM, the conventional feedback loop may be removed. In this case, the reference values, $\tilde{x}[i]$, of Section III-B, are given by $\tilde{x}[i] = x[i-1]$. The proof for CR is essentially the same, however it would require a change in indexing notation to that adopted in Section III-B. DPCM transmitter and receiver schematics are depicted in Fig. 17. In the figure operator, D, represents a delay of one element in the sequence. The decoded sequence is the approximation, $y[i]$, to $x[i]$. We assume that the initial state of the receiver is $y[0] = \alpha_n$ and that the uniform

quantization step size is Δ_n. Then the integer sequence of difference symbols, $\delta\mathcal{Q}_n[i]$, is given by

$$\begin{aligned}\delta\mathcal{Q}_n[i] &= \left\langle \frac{x[i]-y[i-1]}{\Delta_n} \right\rangle \\ &= \left\langle \frac{x[i]-\left(\alpha_n+\Delta_n\sum_{j=1}^{i-1}\delta\mathcal{Q}_n[j]\right)}{\Delta_n} \right\rangle \\ &= \left\langle \frac{x[i]-\alpha_n}{\Delta_n} \right\rangle - \sum_{j=1}^{i-1}\delta\mathcal{Q}_n[j] \end{aligned} \tag{28}$$

where $\langle\cdot\rangle$ denotes rounding to the nearest integer. We define the uniform quantization operator, $\mathcal{Q}_n(\cdot)$, by

$$\mathcal{Q}_n(x) \triangleq \left\langle \frac{x-\alpha_n}{\Delta_n} \right\rangle \tag{29}$$

and obtain

$$\begin{aligned}\delta\mathcal{Q}_n[i] &= \left(\delta\mathcal{Q}_n[i]+\sum_{j=1}^{i-1}\delta\mathcal{Q}_n[j]\right) \\ &\quad - \left(\delta\mathcal{Q}_n[i-1]+\sum_{j=1}^{i-2}\delta\mathcal{Q}_n[j]\right) \\ &= \mathcal{Q}_n(x[i]) - \mathcal{Q}_n(x[i-1])\end{aligned}$$

from (28) and (29). As mentioned, $x[i-1] = \tilde{x}[i]$ for the DPCM case analysed here. We have demonstrated that the DPCM difference sequence may be obtained by first quantizing $x[i]$ and the reference sequence, $\tilde{x}[i]$, with the uniform quantizer of (29) and then taking the difference of these quantized sequences. *The feedback loop is no longer needed.* We also observe, from (28) and (29), that the decoded sequence, $y[i]$, is given by

$$y[i] = \alpha_n + \Delta_n\sum_{j=1}^{i}\delta\mathcal{Q}_n[j] = \alpha_n + \Delta_n\mathcal{Q}_n(x[i]).$$

ACKNOWLEDGMENT

The authors would like to thank J. Liu of Sun Microsystems for his collaboration and his original suggestion of this area of research. We would also like to thank A. Wong of Bellcore for making available the MPEG-1 encoder software cited in this paper.

REFERENCES

[1] E. H. Adelson, E. Simoncelli, and R. Hingorani, "Orthogonal pyramid transforms for image coding," in *Proc. SPIE* (Cambridge, MA), Oct. 1987, pp. 50–58.
[2] V. M. Bove and A. B. Lippman, "Scalable open-architecture television," *SMPTE J.* pp. 2–5, Jan. 1992.
[3] "CCITT SG XV WP/1," *Description of Test Model 3.* Document AVC-400. Nov. 1992.
[4] E. Chang and A. Zakhor, "Scalable video coding using 3-D subband velocity coding and multirate quantization," in *Proc. Int. Conf. Acoust., Speech, Signal Processing* (Minneapolis, MN), vol. 5, Apr. 1993, pp. 574–577.
[5] M. R. Civanlar and A. Puri, "Scalable video coding in frequency domain," in *Proc. SPIE Symp. Visual Commun., Image Processing* (Boston, MA), vol. 1818, pt. 3, Nov. 1992, pp. 1124–1134.
[6] N. Jayant and P. Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.
[7] J. Kovacevic and M. Vetterli, "Nonseparable multidimensional perfect reconstruction banks and wavelet for $\mathcal{R}^n$," *IEEE Trans. Inform. Theory,* vol. 38, no. 2, pp. 533–555, Mar. 1992.
[8] J. Ohm, "Temporal domain sub-band video coding with motion compensation," in *Proc. Int. Conf. Acoust., Speech, Signal Processing* (San Francisco, CA), vol. 3, Mar. 1992, pp. 229–232.
[9] W. Pearlman, "Performance bounds for subband coding," in *Subband Image Coding,* J. Woods, Ed. Norwell, MA: Kluwer, 1990, pp. 1–41.
[10] J. Rissanen and G. Langdon, "Arithmetic coding," *IBM J. Res., Develop.,* vol. 23, no. 2, pp. 149–162, Mar. 1979.
[11] A. Said and W. Pearlman, "Image compression using the spatial-orientation tree," in *Proc. Int. Symp. Circuits, Syst.* (Chicago, IL), vol. 1, May 1993, pp. 279–282.
[12] J. M. Shapiro, "An embedded hierarchical image coder using zerotrees of wavelet coefficients," in *Proc. 3rd Data Compression Conf.* (Snowbird, UT), 1993.
[13] D. Taubman and A. Zakhor, "Orientation adaptive subband coding of images," *IEEE Trans. Image Processing,* vol. 3, no. 4, pp. 421–437, July 1994.
[14] L. Vandendorpe, "Hierarchical transform and subband coding of video signals," *Signal Processing: Image Commun.,* vol. 4, pp. 245–262, 1992.
[15] J. Woods and S. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 34, pp. 1278–1288, Oct. 1986.
[16] Y. Zhang and S. Zafar, "Motion-compensated wavelet transform coding for color video compression," *IEEE Trans. Circuits, Syst., Video Technol.,* vol. 2, no. 3, Sept. 1992.

David Taubman received the B.S. degree in computer science and mathematics in 1986 and the B.E. degree in electrical engineering in 1988, both from the University of Sydney, Australia. He received the M.S. degree in electrical engineering in 1992 from the University of California at Berkeley, where he is currently enrolled in the Ph.D. degree program.

From 1988 to 1990 he worked for the Electricity Commission of N.S.W., Australia. His research interests include subband image and video compression technology.

Mr. Taubman was awarded the University Medal from the University of Sydney in 1988 for Electrical Engineering. In the same year he also received the Institute of Engineers, Australia, Prize and the Texas Instruments Prize for Digital Signal Processing, Australasia.

Avideh Zakhor received the B.S. degree from California Institute of Technology, Pasadena, and the S.M. and Ph.D. degrees from Massachusetts Institute of Technology, Cambridge, all in electrical engineering, in 1983, 1985, and 1987, respectively.

In 1988, she joined the Faculty at the University of California-Berkeley where she is currently Associate Professor in the Department of Electrical Engineering and Computer Sciences. Her research interests are in the general area of signal processing and its applications to images and video and biomedical data. She has been a consultant to a number of industrial organizations and holds four U.S. patents.

Ms. Zakhor was a General Motors scholar from 1982 to 1983, received the Henry Ford Engineering Award and Caltech Prize in 1983, was a Hertz fellow from 1984 to 1988, received the Presidential Young Investigators (PYI) award, IBM junior faculty development award and Analog Devices junior faculty development award in 1990 and Office of Naval Research (ONR) young investigator award in 1992. She is currently Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING and a Member of the Technical Committee for Multidimensional Digital Signal Processing.