Interpolative Multiresolution Coding of Advanced Television with Compatible Subchannels

K. Metin Uz, *Student Member, IEEE,* Martin Vetterli, *Senior Member, IEEE,* and Didier J. LeGall, *Member, IEEE*

Abstract—**A multiresolution representation for video signals is introduced. A three dimensional spatiotemporal pyramid algorithm for high quality compression of advanced television sequences is presented. The scheme utilizes a finite memory structure and is robust to channel errors, provides compatible subchannels, and can handle different scan formats, thus it is well-suited for the broadcast environment. Additional features such as fast random access and reverse playback make it suitable for digital storage as well. Model based processing is used both over space and time, where motion based interpolation is used. Such as interpolation in an FIR scheme solves uncovered area problems, considerably improving the temporal prediction. Complexity is comparable to existing recursive schemes. Computer simulations indicate that high compression factors (about an order of magnitude) is easily achieved with no apparent loss of quality. The scheme also has a number of commonalities with the emerging MPEG standard.**

I. Introduction

THE evolution of the current television standards toward increased quality and realism builds on higher spatial resolution, wider aspect ratio, better chroma resolution, digital audio (CD-quality) and possibly a new scanning format. In addition to the high bandwidth requirements, transmission systems for advanced television face the challenge that the quality has to be maintained throughout the system: for this reason component signals will be preferred over composite and digital representation over analog.

The bandwidth requirements for advanced television (typically more than 1 Gbit/s) ask for powerful digital compression schemes so as to make transmission and storage manageable. The quality requirements and the high resolution of advanced television material make very high signal to noise ratios necessary. It is therefore required to develop source coding schemes for digital video signals which achieve a compression of an order of magnitude or more at the highest possible quality. Two specific cases of interest are contribution quality advanced television at around 100–140 Mbit/s (where objective quality has to be nearly perfect to allow post processing like chroma-keying) and distribution quality for the consumer at rates which are 2–5 times lower and where high subjective quality is required. Besides production and distribution of advanced television (ATV), another application of great interest is the coding for digital storage media (e.g., VTR, CD-ROM), where it is desirable to be able to access any segment of the data, as well as to browse the data (i.e., fast forward or reverse search).

Currently, there is an on-going debate between proponents of interlaced and non-interlaced (also called sequential or progressive) scanning formats for ATV. Both formats have their respective advantages: interlaced scanning saves bandwidth and is well matched to current display and camera technology, and non-interlaced scanning is better suited for graphics and movie applications. In this paper, we will thus deal with both scanning formats.

Our approach to the compression problem of ATV is based on the concept of multiresolution (MR) representation of signals. This concept has emerged as a powerful tool both for representation and for coding purposes. Then, we choose a finite memory (or finite impulse response, FIR) scheme for robustness and for fast random access. The resulting FIR multiresolution scheme successfully addresses the following problems of interest in coding, representation and storage of ATV:

- signal decomposition for compression purposes;
- representation well suited for fast random access or reverse mode in digital storage devices;
- robustness and error recovery;
- suitable signal representation for joint source/channel coding;
- compatibility with lower resolution representations.

Note that multiresolution decomposition is also called hierarchical or pyramidal decomposition, and associated coding schemes are sometimes called embedded or layered coding methods. In the framework of coding, multiresolution decompositions go back to pyramid coding [1] and subband coding [2]–[5], and in applied mathematics, they are related to the theory of wavelets [6], [7]. Note that in this paper, the multiresolution concept will be employed not only for coding but also for motion estimation [8] (as an approximate solution to an optimization problem), a technique also known as hierarchical motion estimation [9], [10].

For robustness purposes, it is advantageous to develop coding schemes with finite memory. This can either be achieved with an inherently finite memory approach, or by periodically restarting a recursive scheme. If a multiresolution decomposition is desired, be it for compatibility purposes or for joint source/channel coding, it turns out that recursive schemes employing a prediction loop, like DPCM or motion compensated hybrid DCT coding [11] have some loss in performance [12]. This is due to the fact that only a suboptimal prediction based on the coarse resolution is possible. In the case of coding for storage media, FIR schemes facilitate easy random access to the data. Additional features such as fast search or reverse playback are

Manuscript received July 1, 1990; revised November 19, 1991.

K. M. Uz's work was supported in part by the National Science Foundation under grant CDR-84-21402. M. Vetterli's work was supported in part by the National Science Foundation under grants CDR-84-21402, MIP-88-08277, MIP-90-14189, and Bell Communications Research.

K. M. Uz and M. Vetterli are with the Department of Electrical Engineering and Center for Telecommunications Research, Columbia University, New York, NY 10027.

D. J. LeGall is with C-Cube Microsystems, 339 W. Trimble Road, San Jose, CA 95131. Part of this author's work was performed while he was with Bell Communications Research, Morristown, NJ 07960.

IEEE Log Number 9042187.

1051-8215/91/0300-0086$01.00 © 1991 IEEE

provided at almost no extra cost. While error correction is widely used in magnetic media, uncorrectable errors are still unavoidable. The finite memory structure of the scheme assures that errors will not be lasting for more than a few frames.

Among the various possible FIR multiresolution schemes, we choose to develop a three dimensional pyramidal coding scheme which uses 2-D spatial interpolation over frames, and motion based interpolation between frames. Note that the temporal interpolation is similar to the proposed MPEG standard [13]. We will justify the reasons for this choice by examining pros and cons in detail, and showing that the resulting coding scheme marries simplicity and compression at high quality. The complexity of the overall scheme is comparable to alternative coding schemes that are considered for high quality ATV applications.

The outline of the paper is as follows. We begin by looking at multiresolution representations for coding, and examine two representative techniques in detail: subband and pyramid coding. We compare these two cases in terms of representation, coding efficiency, and quantization noise. Section IV describes the spatiotemporal pyramid, a three-dimensional pyramid structure that forms the basis of our coding scheme. In the following section, we focus on the temporal interpolation within the pyramid, describing the multiresolution motion estimation and motion based interpolation procedures in detail. The coding system and simulation results are given in section VI, along with a discussion of relation to the evolving MPEG standard [13]. Finally, we analyze the computational and memory complexity of the proposed scheme.

II. Multiresolution Signal Representations for Coding

The idea of multiresolution is similar to that of successive approximation. A coarse approximation to a signal is refined step by step, until the signal is obtained at the desired resolution. Very similarly, an initial solution to an optimization problem can be refined stepwise, until the full resolution solution is achieved. To get the coarse approximation, as well as to refine this approximation to increase the resolution, one needs operators adapted to the particular resolution change. These can be linear filters, or more sophisticated model based operators. Typical examples are decimation of a signal (fine-to-coarse), which is usually preceded by a lowpass anti-aliasing filter, and upsampling (coarse-to-fine) which is followed by an interpolation filter. We will see that video processing calls for more sophisticated, nonlinear operators, such as motion based frame interpolation used to increase time resolution.

Multiresolution approaches are particularly successful when some a priori structure or hierarchy can be found in the problem. A classic approximation technique used in statistical signal processing and waveform coding is the Karhunen–Loeve transform (KLT) [14]. Given a vector process (typically obtained by blocking a stream of samples), one computes a linear transform $\boldsymbol{T}$ such that $y_n = \boldsymbol{T} \cdot \boldsymbol{x}_n$. The rows of the transform matrix are chosen as the eigenvectors of the autocorrelation matrix $\boldsymbol{R}$ of $\boldsymbol{x}_n$ which is symmetric (by stationarity assumption), and therefore $\boldsymbol{T}$ is unitary. The samples of y_n are thus decorrelated. Now, the rows of $\boldsymbol{T}$ can be ordered so that:

$$E[y_n(i)^2] \geq E[y_n(j)^2], \qquad i < j. \tag{1}$$

That is, the first k coefficients of the KLT are a best k coefficient approximation to the process x_n in the mean squared error (MSE) sense.

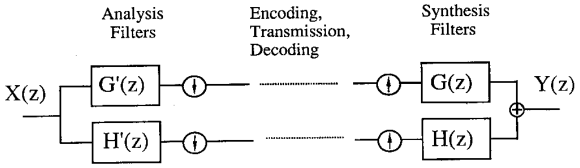


Fig. 1. Two channel subband coding system.

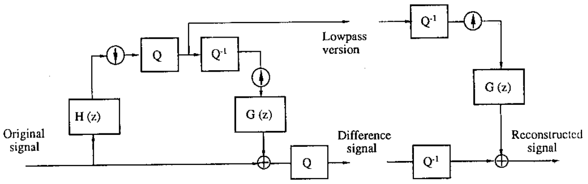


Fig. 2. One step in pyramid decomposition and coding. Note that only source of reconstruction error is quantizer for difference signal.

Transform and subband coding, two successful standard image compression techniques when combined with appropriate quantization and entropy coding, can be seen as variations on this idea. In terms of multiresolution approximation, their suitability stems from the fact that images typically have a power spectrum that falls off at higher frequencies. Thus, low frequency components of a transform coded picture form a good approximation to the picture. This multiresolution nature of transform coding is used for example in progressive transmission of pictures. Subband coding (see Fig. 1) can be seen as a transform with basis vectors extending over more than one block. Constraints are imposed on the analysis and synthesis filterbanks so as to achieve perfect reconstruction [15]. Note that both transform and subband decompositions form a one-to-one mapping, as the number of samples is preserved. In contrast, pyramid decomposition is a redundant representation, since a low resolution version as well as a full resolution difference are derived (see Fig. 2). This redundancy, or increase in the number of sample points, becomes negligible as the dimensionality increases, and allows greater freedom in the filter design.

III. Subband and Pyramid Coding

Transform coding and subband coding are very similar decomposition methods, and will thus be discussed together. Below, we will compare and contrast respective advantages and problems of subband schemes versus pyramid schemes. It should be noted that subband decomposition can be viewed as a special case of pyramid decomposition using constrained linear operators in the absence of quantization [16], [7].

A. Characteristics of Subband Schemes

The most obvious advantage of subband schemes is the fact that they are critically sampled, that is, there is no increase in the number of samples. The price paid is a constrained filter design and therefore a relatively poor lowpass version as a coarse approximation. This is undesirable if the coarse version is used for viewing in a compatible subchannel or in the case of progressive transmission. Only linear processing is possible in subband coding systems, and any nonlinearity has effects which are hard to predict. In particular, quantization noise can produce artifacts in the reconstructed signal which are difficult to foresee.

The problems are linked to the fact that the bound on the maximum error produced in the reconstructed signal because of quantization in the subbands is fairly weak. This is different from the MSE (or l_2 norm of the error), which is conserved since the subband decomposition and reconstruction processes are unitary operators[1]. However, the maximum error is given by the l_∞ norm, which is not conserved by unitary transforms. To make this point clear, let us consider the case of the size N DCT. The first row of the IDCT is equal to $1/\sqrt{N}$ [1, 1, . . . , 1]. If the vector of quantization errors is colinear with this row, the backtransformed vector of errors is equal to $[\sqrt{N}\delta, 0, \ldots, 0]$ (where δ is the quantization step in the transform domain). Thus, all the errors accumulated on the first coefficient! Note that $\sqrt{N}$ is typically equal to 8 (corresponding to blocks of 8×8 pels) in image coding applications. Subband schemes behave similarly.

Three dimensional subband coding has been proposed for video coding [18], [19], but has not been able to compete with motion-compensated coding methods in terms of compression efficiency. It seems difficult to naturally include motion compensation within a subband scheme. Motion is a time domain concept, while a subband decomposition leads to a (partly) frequency domain description of the sequence. Thus, motion compensation can be done outside the subband decomposition [20] (in which case SBC replaces DCT for encoding the prediction error) or can be seen as preprocessing [21] (where only block transforms are used over time). Motion compensation within the bands suffers from accuracy problems [22], and from the fact that independent errors can accumulate after reconstruction [23].

B. Pyramid Coding

We have seen that pyramid decomposition is a redundant representation. This redundancy, or increase in the number of sample points, becomes negligible as the dimensionality increases. In a one dimensional system, the increase is upper-bounded by $1 + 1/2 + 1/4 + \ldots < 2$, in two dimensions by $1 + 1/4 + 1/16 + \ldots < 4/3$ and in three dimensions, by $1 + 1/8 + 1/64 \ldots < 8/7$. That is, in the three dimensional case that we will be using, the overhead is less than 15%. At the price of this slight oversampling, one gains complete freedom in the design of the coarse-to-fine and fine-to-coarse resolution change operators, which can be matched to the signal model, and can be nonlinear [24]. Constraints in the transform or subband decompositions often result in compromises in the filter quality. If linear filters are chosen to derive the lowpass approximation in a pyramid, it is possible to take very good lowpass filters, and derive visually pleasing coarse versions. Therefore, pyramids can be a better choice when high visual quality must be maintained across a number of scales [25]. We also observe that the l_∞ problem in transform and subband coding case can be avoided in pyramids by quantization noise feedback. A detailed analysis follows in the next subsection.

C. Analysis of Quantization Noise

In this section, we analyze the propagation of quantization noise in pyramid and subband decomposition schemes. We will consider three representative cases: an iterated two-channel subband splitting, and a pyramid with and without error feedback. Simulations are based on actual data from the well known image *Lenna*.

A three stage subband analysis bank and the corresponding synthesis bank are depicted in Figs. 3 and 4. We assume each band is independently quantized by a scalar quantizer (band quantizer) of equal step size and that quantization noise can be modeled as white. Furthermore, we assume a similar quantizer (with a finer step size) is used following each filter, to model the finite wordlength effects (internal quantizer). For simplicity, we focus on the synthesis bank, although a similar conclusion can be reached for the analysis bank. Let q_i be the noise due to the quantizer for band i, and f_i be the internal quantizer noise due to the synthesis filter pair. Then $E_i(z)$, the z transform of e_i can be expressed as

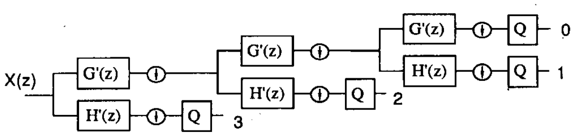


Fig. 3. Subband analysis filterbank where band splitting has been iterated three times on low band, yielding for channels with logarithmic spacing.

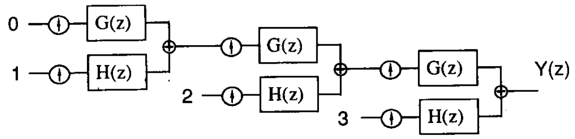


Fig. 4. Corresponding subband synthesis filterbank.

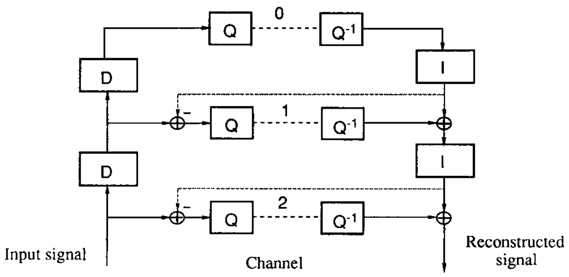


Fig. 5. Three level pyramid coding, with feed-back of quantization of high layers into prediction of lower ones. D and I stand for decimation and interpolation operations. Only one source of quantization error remains, namely, that of the highest resolution difference signal.

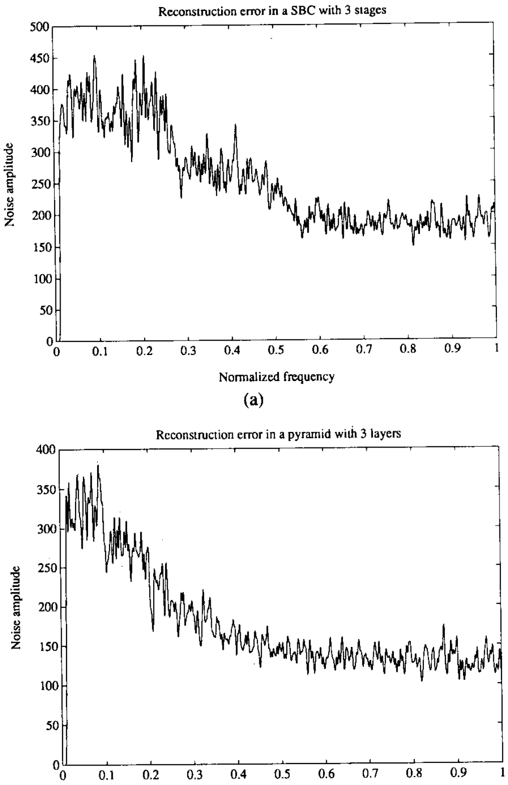
$$E_i(z) = E_{i-1}(z^2)G(z) + Q_i(z^2)H(z) + F_i(z). \quad (2)$$

(Upsampling by 2 means replacing z by z^2 in the z transform domain [17]) which leads to a final reconstruction error

$$E_N(z) = Q_N(z)\prod_{i=0}^{N-1} G(z^{2^i}) + \sum_{i=1}^{N-1}\left(\prod_{n=1}^{i} G(z^{2^n})\right) \cdot \left\{Q_i(z^{2^i})H(z^{2^i}) + F_i(z^{2^i})\right\}. \quad (3)$$

We note that the noise spectrum is biased into low frequencies. For an intuitive explanation, consider the signal flow graph corresponding to the analysis-synthesis system. Band N, the subsignal that takes the longest path ($2N$ lowpass filters) covers only $1/2^N$ of the spectrum at the dc end. Therefore, more finite wordlength effects are visible at low frequencies. A numerical simulation was done with the 32-tap Johnston QMF filter pair (type C) [26], using 10 bits for the internal quantizers, and 6 bits for the band quantizers. The resulting reconstruction error spectrum is depicted in Fig. 6(a). We should note that in practice, one would choose finer quantizers for the lower bands, partially alleviating the problem, although the accumulation due to finite wordlength effects is unavoidable.

[1] Actually, this is only true for paraunitary filter banks [17], but holds approximately true for most perfect reconstruction filter banks.



(a)

(b)

Fig. 6. Reconstruction error power spectrum: (a) SBC with three stages. (b) Pyramid with three layers without quantization error feedback.

Next, we consider an N stage pyramid without quantization error feedback. We assume that each stage employs a linear scalar quantizer, and that the quantization noise can be modeled as white. For simplicity, we assume that quantizers have equal step size. Let q_i be the noise due to the quantizer at layer i, defined such that coarsest layer is layer 0, and let the reconstruction error at layer i be denoted by e_i. For a linear filter $H(z)$, we can easily analyze the response of the system to the quantization noise by assuming the only input to the decoder is the set $\{q_0, q_1, \ldots, q_N\}$. Then

$$E_i(z) = E_{i-1}(z^2)H(z) + Q_i(z) \tag{4}$$

which, by iteration, gives the final reconstruction error as

$$E_N(z) = \sum_{i=0}^{N} \left\{ \prod_n H(z^{2^n}) \right\} Q_i(z^{2^i}). \tag{5}$$

Qualitatively, it is easy to see how the quantization noise propagates across the layers: The initial error $e_0 = q_0$ is white. Upsampling creates a replica in the spectrum, and h, typically a lowpass filter, attenuates the replica. Thus, e_1 consists of white q_1 plus this lowpass noise. At each layer, previous reconstruction error is squeezed into an (approximately) halfband signal, and white noise is added. The results of a numerical simulation using three stages is shown in Fig. 6(b). Here the filters are those used by Burt and Adelson [1], where $a = 0.6$, and the quantizer step size is 4.

Notice the quantization error is hard to control in both cases. Furthermore, the error spectrum is biased toward low frequencies, particularly undesirable since the human visual system (HVS) is much more sensitive to lower frequencies [27]. For comparison, in a pyramid with feedback the only source of quantization error is the final quantizer, since prior to errors are corrected at each layer. We can thus guarantee a bound δ on the maximum error, and also tailor the quantization to the HVS, by shaping the error spectrum and by using masking based on the original sequence.

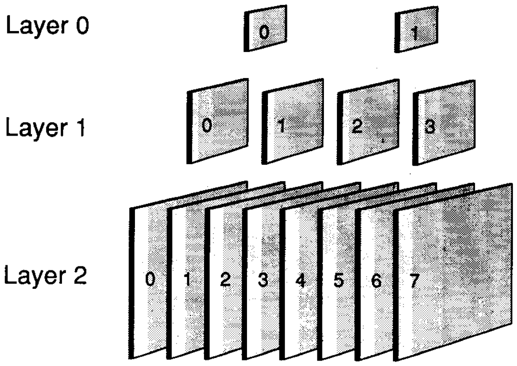


Fig. 7. Spatiotemporal pyramid structure.

IV. The Spatiotemporal Pyramid

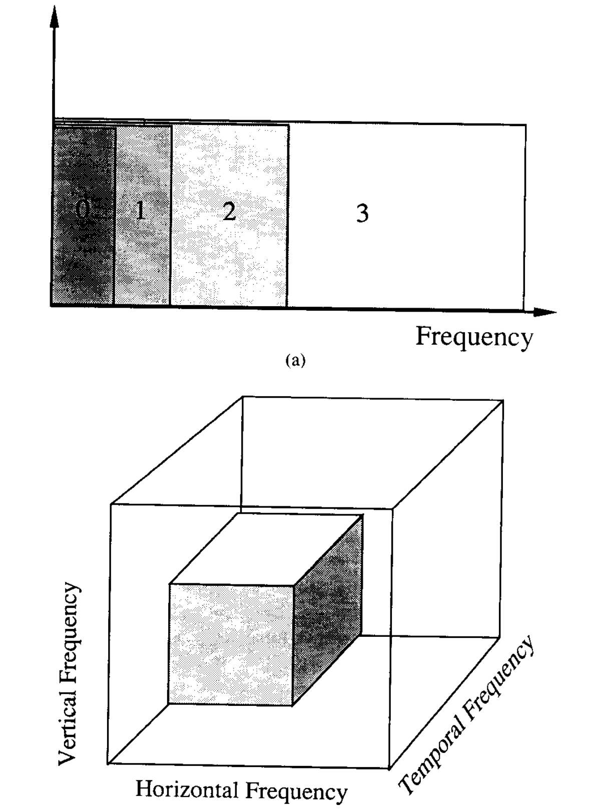
In this section, we introduce the spatiotemporal pyramid, a multiscale representation of the video signal. It consists of a hierarchy of video signals at increasing temporal and spatial resolutions (see Fig. 7). Here, we should stress that the video signal is not a true 3-D signal, but can be modeled as a 2-D signal with an associated displacement field. This fundamental difference between space and time is taken into account in the pyramid by the choice of motion based processing over time. This also justifies the lack of filtering prior to temporal decimation [28].

The structure is formed in a bottom-up manner, starting from the finest resolution, and obtaining a hierarchy of lower resolution versions. Spatially, images are subsampled after anti-aliasing filtering. Temporally, the reduction is achieved by simple frame skipping.

The frequency division obtained with a pyramid is depicted in Fig. 8(b). The decomposition provides a logarithmic scaling in frequency (see Fig. 8(a)), with the bandwidth increasing by a factor of 2 in each dimension down the pyramid. Thus, the spectral volume of the signal is increased by a factor of 8 at each coarse-to-fine resolution change (actual scaling factor may depend on the sampling grid).

The encoding is done in a stepwise fashion, starting at the top layer and working down the pyramid in a series of successive refinement steps. At each step, the signal is first spatially interpolated, increasing the spatial resolution by 2 in each dimension (a factor of 4 in the number of samples per frame). Motion based interpolation follows, doubling the temporal resolution and completing the reconstruction of the next layer. We describe the motion-based processing in more detail in the next section, and now focus on some key properties of the scheme.

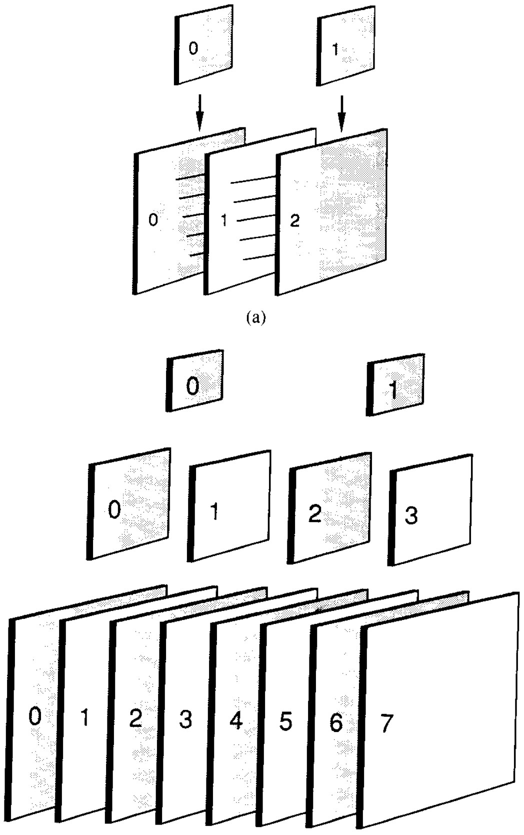
The structure forms the basis of a finite-memory coding procedure. The frames at a particular layer are based upon the frames directly above them. Note that the dependence graph is in the form of a binary tree, in which the nodes are the individual



(a)

(b)

Fig. 8. Logarithmic frequency division in pyramid. (a) One dimensional pyramid with four layers. (b) Three dimensional pyramid with two layers. Notice that the spectral volume doubles in the first case, but increases eight-fold in the latter.



(a)

(b)

Fig. 9. Reconstruction of pyramid. (a) One step of coarse-to-fine scale change (b) The reconstruction pyramid. Note that approximately one half of frames in structure (shown as shaded) are spatially coded/interpolated.

frames, and the leaves are the frames in the final layer. Therefore any channel error has a finite (and short) duration effect on the output: only the frames with branches passing through the error are affected. With typical parameters, this implies no error can last more than a small fraction of a second. This is in contrast with predictive coders, where the prediction loop has to be restarted frequently to avoid error accumulation.

The encoding procedure is also computationally attractive. Although the scheme provides a 3-D decomposition, the complexity is kept linear by using separable kernels for interpolation. We should note that an optimal algorithm would require complex motion-adaptive spatiotemporal filters, computationally expensive and hard to design. By decoupling space and time, we achieve significant reductions in computation with little penalty, especially in view of our source model: A sequence is formed by images displaced smoothly over time.

The coarse-to-fine scale change step is illustrated in Fig. 9 (a). First the spatial resolution is increased, then the temporal interpolation is done based on these new frames at the finer scale. We should note that reversing the order of these operations would cause increased energy in the difference signals to be encoded. In other words, interpolation is statistically more successful over time than over space, so the temporal difference signal has lower energy and thus is easier to compress.

As a side note, we should note that this mismatch problem can be partially alleviated by interpolating more than one frame over time[2]. However, this scheme is not without drawbacks: It becomes harder to maintain the quality between frames, and the effect of a channel error has now longer duration.

We have seen that motion based interpolation is central to the multiresolution approach over time. Therefore we will focus on the motion estimation problem in the next section.

V. Multiresolution Motion Estimation and Interpolation

Motion estimation is based on a multiresolution algorithm: The motion field is initially estimated on a coarse grid, and successively refined until a sufficiently dense motion field is obtained. Here, we are concerned with computing the apparent 2-D motion, rather than the actual 3-D velocities. However, the motion based interpolation still requires a reasonably accurate estimate, not just a match in the MSE sense.

Hierarchical approaches have been applied to motion estimation problem [9], [29], [30]. The motivation for the algorithm lies in the observation that typical scenes frequently contain motion at all scales. Camera movements such as pan and zoom induce global motion, and objects in the scene move with velocities roughly proportional to their sizes.

The structural model, i.e., the relation between the image and

[2] Indeed, the current MPEG proposal [13] for video coding provides two interpolated frames between conventional predicted frames.

motion field, also suggests a MR strategy. Consider a scene consisting of a superposition of sine waves (any frequency domain technique implicitly assumes this). Now consider uniformly displacing this scene in time. Looking at a single sinusoid, the largest displacement that can be computed is limited to less than half its wavelength. With low frequencies, it's hard to resolve small differences in displacement, i.e., precision is reduced. With high spatial frequencies, one gets high resolution at small displacements, but large displacements are ambiguous at best, due to aliasing. However, we assume that all these components move at the same velocity, because they belong to the same rigid object. So a coarse-to-fine strategy at estimating motion seems to be a natural choice: Start at a low resolution image, compute a coarse motion field, refine the motion estimate while looking at higher spatial frequencies.

The second argument in favor of a MR technique is the computational complexity. Brute force algorithms such as full search require $O(d^2)$ searches, where d is the maximum allowable displacement, and is typically a fraction of the picture size. Thus, as the picture definition increases, so does d, quadratically increasing the search complexity. In contrast, MR schemes can be designed with roughly logarithmic complexity. So, the MR choice is also a computationally attractive one.

An inherent difficulty in motion compensation is the problem of covered/uncovered areas. In a predictive scheme, one cannot account for the area that has just been uncovered: an object appears on screen for which no motion vector can be computed. Interpolation within an FIR structure elegantly solves this problem: covered areas are visible in the past, and uncovered areas in the future.

We will present the motion estimation algorithm in two steps. First, we describe a hierarchical symmetric mode search similar to that of Bierling [10], but uses the previous and the following frames to compute the motion field. Next, we consider the problem of motion based interpolation, and modify the estimation algorithm for the best reconstruction. Essentially, the algorithm consists of three concurrent searches, and a control mechanism to select the best model. In effect, this also selects the interpolation mode, and sends it as side information.

A. Basic Search Procedure

We start with the video signal $I(r, n)$ where r denotes the spatial coordinate (x, y), and n denotes the time. The goal is to find a mapping $d(r, n)$ that would help reconstruct $I(r, n)$ from $I(r, n-1)$ and $I(r, n+1)$. We assume a restrictive motion model, where the image is assumed to be composed of rigid objects in translational motion on a plane:

$$I(r, n) = I\big(r - d(r, n), n - 1\big). \tag{6}$$

We also expect homogeneity in time, i.e.,

$$I(r, n) = I\big(r + d(r, n), n + 1\big). \tag{7}$$

Furthermore, we are using a block based scheme, expecting these assumptions are approximately valid for all points within a block b using the same displacement vector d_b. These assumptions are easily justified when the blocks are much smaller than the objects, and temporal sampling is sufficiently dense (we have used 8 × 8 blocks, but any size down to 4 × 4 works quite well).

In what follows, we change the notation slightly, omitting the spatial coordinate r when the meaning is clear, and replacing $I(r, n)$ by $I_K(n)$, and $d(r, n)$ by $d_K(n)$.

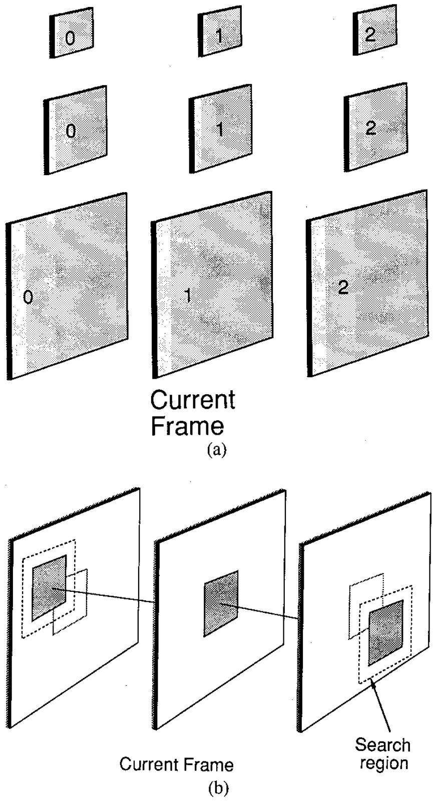


Fig. 10. Motion estimation starts from coarse image, and gradually refines estimate. Search is performed in symmetric fashion within window around previous estimate.

To compute $d_K(n)$, we require a hierarchy of lower spatial resolution representations of $I_K(n)$ denoted $I_k(n)$, $0 \geq k < K$. $I_k(n)$ is computed from $I_{k+1}(n)$ by first spatially low-pass filtering with half-band filters, reducing the spatial resolution. This filtered image is then decimated, giving $I_k(n)$. Note that this reduces by 4 the number of pels in a frame at each level (see Fig. 10).

The search starts at the top level of the spatial hierarchy, $I_0(n)$. The image $I_k(n)$ is partitioned into non-overlapping blocks of size $M \times M$. For every block b in the image $I_k(n)$, a displacement d_b is searched to minimize the matching criterion

$$\sum_{r \in b} | I_k(r, n) - \tilde{I}_k(r, n) | \tag{8}$$

where $\tilde{I}_k(r, n)$ is the motion based estimate of $I_k(n)$ computed as

$$\tilde{I}_k(r, n) = 1/2\big(I_k(r - d_b, n - 1) + I_k(r + d_b, n - 1)\big). \tag{9}$$

Notice that this estimate implies that if a block has moved the distance d between the previous and the current frame, it is expected to move d between the current and the following frame. This constitutes the symmetric block based search.

B. Stepwise Refinement

Going to step down the hierarchy, we want to compute $d_k(n)$, given that $d_{k-1}(n)$ is already known. We may think of d_k as a sampled version of an underlying continuous displacement field.

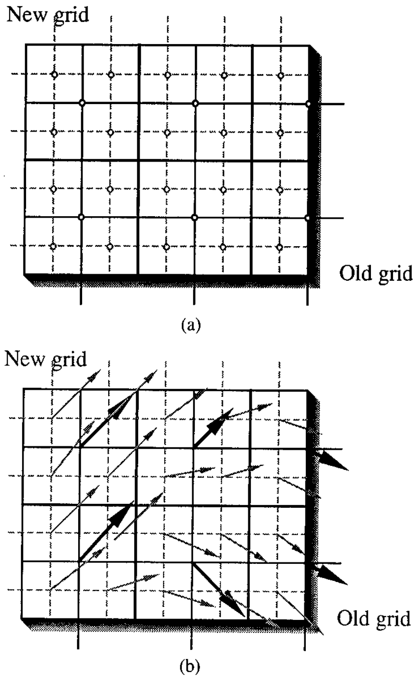


Fig. 11. Stepwise refinement for motion estimation. (a) Blocks and corresponding grids at two successive scales in the spatial hierarchy. (b) Motion field is resampled in new finer grid, giving initial estimate for next search.

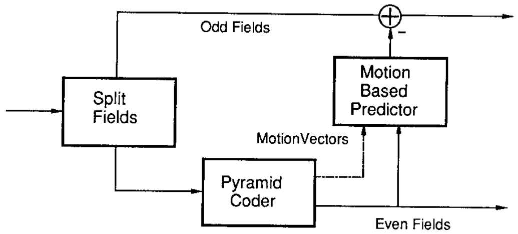


Fig. 12. Block diagram for coding interlaced input signals.

Then $\{d_k : 0 \leq k \leq K\}$ is a set of multi-resolution samples taken from the underlying displacement field. Therefore, d_k can be written as

$$d_k(r, n) = 2\tilde{d}_{k-1}(r, n) + \Delta d_k(r, n) \tag{10}$$

where $\tilde{d}_{k-1}$ is the displacement field d_{k-1} interpolated at the new sampling grid (see Fig. 11), and Δd_k is the correction term for the displacement due to increased resolution at level k. The rescaling by 2 arises due to the finer sampling grid for I_k: the distance between two pels in the upper level has now doubled. Thus, we have reduced the problem of computing d_k into two subtasks which we now describe.

Computing $\tilde{d}_{k-1}$ on the new sampling grid requires a resampling, or interpolation (this may seem to be a minor point, however, the interpolation is crucial for best performance with the constrained search). The discontinuities in the displacement field are often due to the occluding boundaries of moving objects. Therefore, the interpolation may use a (tentative) segmentation of the frame based on the displacement field, or even on the successive frame difference. The segmentation would allow classifying the blocks as belonging to distinct objects, and preventing the corona effect around the moving objects. However, we currently use a simple bilinear interpolation for efficiency, moving the burden to computing Δd_k.

After the interpolation, every block b has an initial displacement estimate $\hat{d}_b$, which is equal to $2\tilde{d}_{k-1}(r_b, n)$ where r_b is the coordinate of the center of block b. Now the search for Δd_k is done inside a window centered around $\hat{d}_b$, i.e.,

$$d_b = \hat{d}_b + \Delta d, \quad \Delta d \in \{(x, y): -D \leq x, y \leq +D\}. \tag{11}$$

Note that the number of displacements searched for a block is constant, while the number of searches increases geometrically down the hierarchy. The procedure is repeated until one obtains the motion field $d_K(n)$, corresponding to $I_K(n)$.

Suppose the displacement in the original sequence $I_K(n)$ were limited to $\pm d_{\max}$ pels in each dimension. Then the displacement in the $(K - k)$th level is limited to $\pm d_{\max}/2^k$, since the frames at this level have been k times spatially decimated by 2. In general, K can be chosen such that the displacement at the top level is known to be less than $\pm D$. Given the choice, we can limit the search for each $d_k(r, n)$ to $\{(x, y): -D \leq x, y \leq +D\}$. This results in $(2D + 1)^2$ tests to compute d_b for each block. The maximum displacement that can be handled is

$$\sum_{i=0}^{K} 2^i D = (2^{K+1} - 1)D. \tag{12}$$

For a typical sequence, D can be 2 or 3, and K can be 3, allowing a maximum displacement of 30 or 45. This constrained window symmetric search algorithm yields a smooth motion field with high temporal and spatial correlation, at the same time allowing large displacements. Three stages in the estimation procedure are shown in Fig. 14.

C. Motion Based Interpolation

Given frames $I(n - 1)$ and $I(n + 1)$, and the displacement $d(n)$, we form $\tilde{I}(n)$, the motion based estimate of $I(n)$ by displaced averaging:

$$\tilde{I}(r, n) = 1/2(I(r - d, n - 1) + I(r + d, n + 1)). \tag{13}$$

Here, we use $d - d_b$, the displacement of the block containing r, i.e., we operate on a block level. There are other alternatives, including a pel-level resampling of the displacement field $d(r)$. However, this would significantly increase the complexity of the decoder, which is kept to a minimum by the blockwise averaging scheme. Furthermore, simulations indicate that the time-varying texture distortions caused by pel-level interpolation are visually unpleasant. It is desirable to preserve textures, but it is especially critical to avoid time-varying distortions which tend to be perceptually unacceptable.

We shall use motion based interpolation as the temporal interpolator in a three dimensional pyramid scheme. However, the method, as presented, has some limitations especially when temporal sampling rate is reduced. Consider temporally decimating $I_1(n)$, call it $I_1^1(n)$ (recall that $I_1(n)$ has a reduced spatial resolution). If the original sequence $I_K(n)$ has a frame-to-frame maximum displacement of $d_{\max}$, this property is preserved in $I_1^1(n)$. However, as we go up the spatial hierarchy, the displacements start to deviate significantly from our original assumptions: We can no longer assume that the velocities are constant, nor that time is homogeneous, although we know that the maximum displacement is preserved. To make matters even worse, the area covered (or uncovered) by a moving object increases proportional to the temporal decimation factor. Thus,

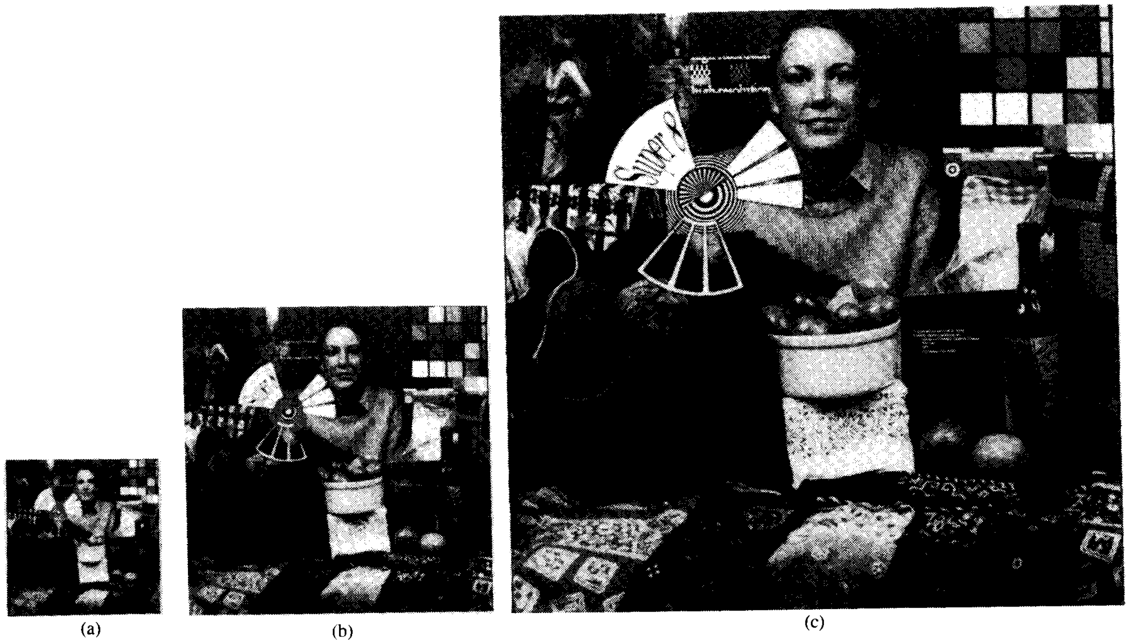


Fig. 13. (a)-(c) Original frame and two coarser versions from MIT sequence (also see Fig. 7). (For color supplement see page 158.)

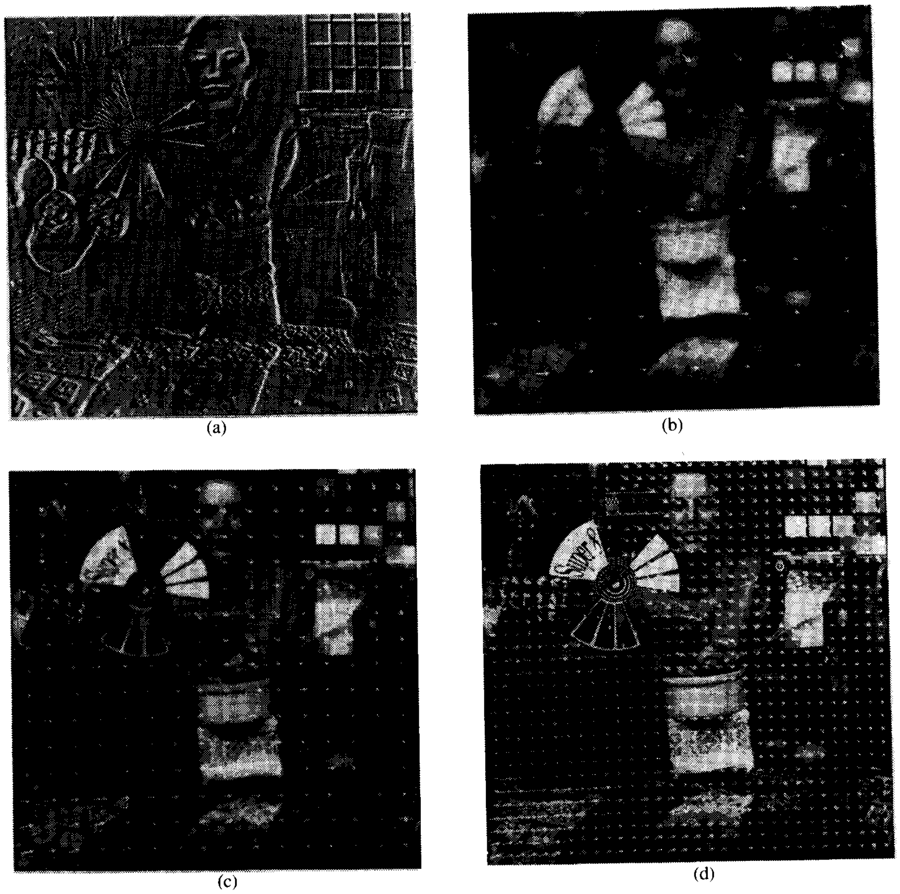


Fig. 14. Motion estimation based on multiresolution search. (a) Successive frame difference, showing considerable amount of motion is present in sequence. (b)-(d) Three stages in motion estimation algorithm. (For color supplement see page 158.)

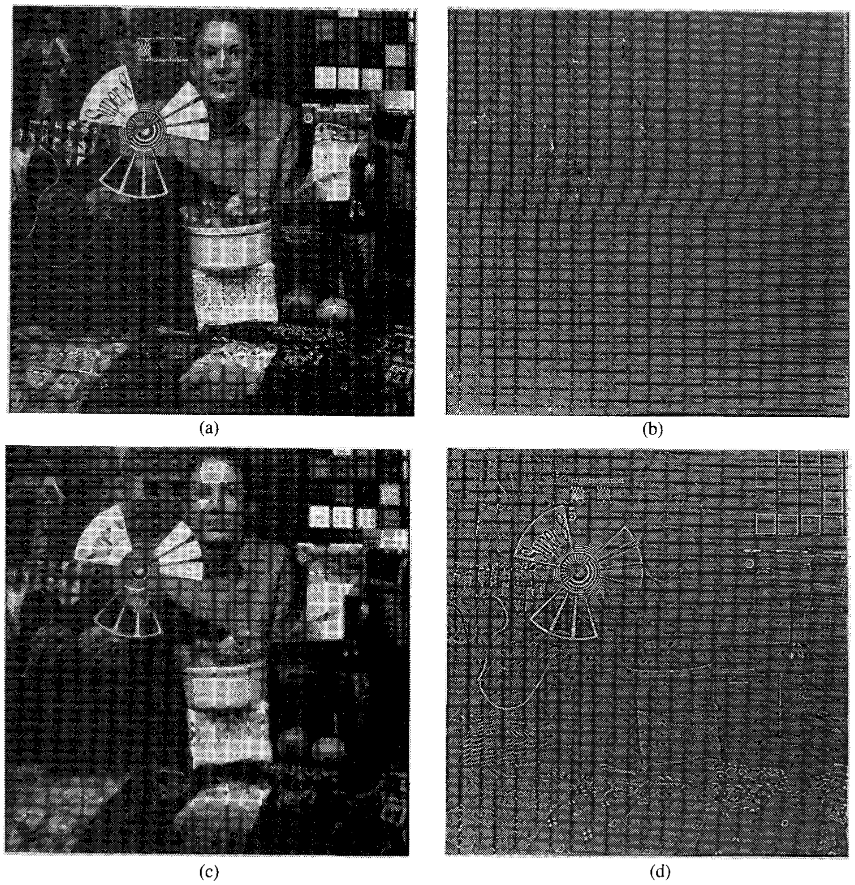


Fig. 15. Interpolated frames and corresponding difference signals. (a) Temporally interpolated frame. Some blocking artifacts can be noticed around wheel. (b) Temporal difference signal to be encoded. (c) Spatially interpolated frame. Note all sharp edges are blurred. (d) Spatial interpolation error to be encoded. (For color supplement of Fig. 15(a) and (c) see page 159.)

an increasingly larger ratio of the viewing area is covered/uncovered as we go up in our pyramid structure. In that case, simple averaging by a symmetric motion vector will likely result in double images and other unpleasant artifacts.

For a successful interpolation, we have to be able to 1) handle discontinuities in motion; 2) interpolate uncovered areas. These are fundamental problems in any motion based scheme; a manifestation of the fact that we are dealing with solid objects in 3-D space, of which we have incomplete observations in the form of a 2-D projection on the focal plane.

To overcome these difficulties, we allow the interpolator to selectively use previous, following, or both frames on a block by block basis. The algorithm is thus modified in the final step, where Δd_K is computed to yield the final displacement d_K. In addition to the symmetric search so far discussed, two other searchers are run in parallel: one using the current and previous, and the other using the following and current frames. Then the motion interpolated blocks using the three schemes are compared against the original block. Now the displacement and interpolation mode minimizing the interpolation error is chosen as the final displacement. (In practice, one may weight the errors, favoring the displacement obtained by the symmetric search). Tests performed using different scenes indicate that the scheme works as intended: Symmetric mode is chosen most of the time, with the other two modes being used 1) when there is a pan, in which case the covered/uncovered image boundary is interpolated from future/past; 2) around moving object boundaries (for the same reason); 3) when there is irregular motion, i.e., the symmetry assumption no longer holds (see Fig. 16). This interpolation technique has originated from an earlier study for MPEG [31], and is similar to the current MPEG proposal [13].

VI. Compression for ATV

Compression of ATV signals for digital transmission is a challenging problem. High quality, meaning both high signal-to-noise ratio and near perfect perceptual quality must be maintained at an order of magnitude reduction of bandwidth. Additional constraints such as cost, real-time implementation and suitability to broadcast environment narrow down the number of alternatives even further.

The layers in the pyramid, except for the top one, consist of interpolation errors, i.e., they are mostly bandpass signals with logarithmic spacing. The bands can be modeled to be independent to a good approximation, although this assumption is often violated around edges in the image. Nevertheless, multiresolution decomposition provides a good compromise between efficiency and coding complexity, and facilitates joint source-channel coding. Properties of the HVS can also be exploited in such a decomposition. HVS has been modeled to consist of independent band-pass channels (at least for still images) [27], [32], and distributing error into these bands provides high compression

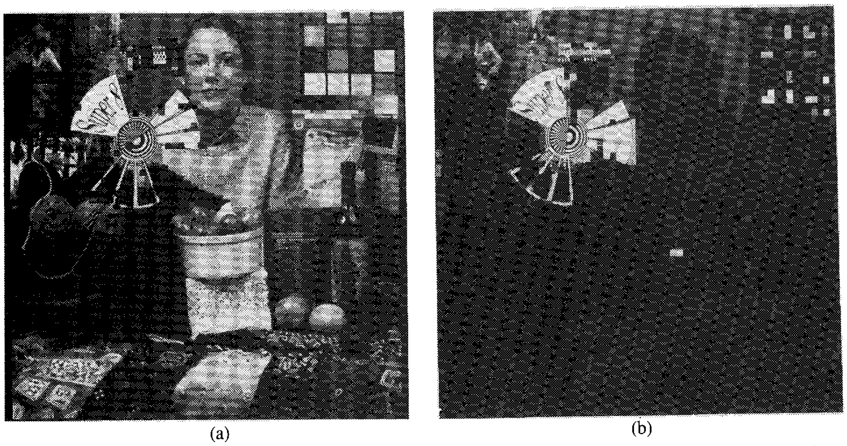


Fig. 16. Temporal interpolation mode. (a) Typical frame. Majority of the blocks are averaged. (b) Motion changes from pan top zoom. Backward or forward mode is used for most blocks, accounting for the fact that motion is highly irregular. (For color supplement see page 159.)

with little unpleasant artifacts. Temporal and spatial masking phenomena [33] can also be used to advantage while maintaining high perceptual quality.

In this section, we apply the multiresolution concepts so far developed to coding for ATV. First we describe how the scheme can be applied for coding interlaced sequences, and then give some simulation results. We show that both scan types can be successfully merged within a multiresolution hierarchy, providing a versatile, compatible representation for the video signal.

A. Compatibility and Scan Formats

All existing broadcast TV standards currently employ interlaced scan, and we can expect it to dominate the ATV scene in the foreseeable future. The inherent bandwidth reduction and efficient utilization has been the major factor in its acceptance. Today's high resolution, low noise cameras rely on interlacing: switching to non-interlated scan may degrade the performance by as much as 9 dB [34] (for the same number of lines). On the other hand, non-interlaced scan has many desirable features, including less visible artifacts (absence of line flicker), and convenience for digital processing. Furthermore, some sources such as movies and computer generated graphics are available only in non-interlaced format. Both objective and subjective tests indicate the non-interlaced displays are superior [35]. The next generation systems may be required to handle both scan types, which we show can actually be achieved within a multiresolution scheme at little extra cost. An excellent overview of sampling and reconstruction of multidimensional signals can be found in [36].

Approaches to the coding of interlaced television signals have been studied within CCIR's CMTT/2 Expert Group [37], [38]. The CMTT algorithm considers a recursive coder with three prediction modes (intrafield, interfield, or motion-compensated interframe). The work done in the CMTT/2 Expert Group did not consider interpolative or non-causal prediction, however, the idea of getting a predictor—forward or backward—from either the nearest reference field or the nearest cosited field is critical in dealing with interlaced video: depending on the amount and the nature of motion either field may be a good candidate for prediction.

In a finite memory scheme, we are faced with a choice: either group two adjacent fields to form a reference frame, or limit the references to one field. The second solution is much more natural in a temporal pyramid. If the decimation factor is two, the low temporal resolution signal is all of parity and the signal to be interpolated is of the other parity. Work along those lines has also been performed independently by Wang and Anastassiou and is reported in [39]. In a temporal pyramid with a decimation factor of two, the input signal is thus separated into odd and even fields, as illustrated in Fig. 12. The three-dimensional pyramid decomposition is performed as described on the even fields. As also suggested in [39], the odd fields are encoded by motion-compensated interfield interpolation. The three interpolation modes previously described can again be used on a block by block basis.

Overhead can be kept low by using the motion information already available at the decoder, or a new temporal interpolation step can be performed, along with transmission of motion vectors and interpolation modes. This is a particularly attractive solution, for it marries the simplicity of non-interlaced sampling inside the spatiotemporal pyramid with the scan compatibility by providing an interlaced signal at the finest resolution. Initial results in this direction look promising, with compression and quality comparable to those obtained with non-interlaced sequences.

B. Results

The proposed coding system consists of an entropy coder following the three-dimensional pyramidal decomposition. A discrete cosine transform (DCT) based coder is used to encode the top layer and the subsequent bandpass difference images. The coding algorithm is similar to the JPEG standard [38], [39] for coding still images. DCT is probably not the best choice for coding difference signals, as it approximates the KLT for stationary signals with high autocorrelation [14]. However, it has been widely used in practice, and VLSI implementations make it an attractive choice for inexpensive real-time implementations.

There are several parameters for adjusting the quality versus bandwidth. Each DCT coefficient is quantized by a linear scalar quantizer. Luminance and chrominance components have separate quantizers, with steps adjusted based on MSE and perceptual quality. Typically, chrominance components use about 20% of the total bandwidth for MSE comparable to the luminance MSE. The bit allocation among the layers is determined by setting the quantizer step size to maintain good visual quality in each layer. Selection of optimal quantizers across the multireso-

TABLE I

Layer	Spatial	Temporal	Overall
1	3.06	N/A	0.04
2	2.10	0.55	0.16
3	2.21	0.45	1.33
Total	2.52	0.52	1.53

Bit allocation to various layers for the MIT sequence. All numbers are in bits per pixel. The overall column indicates the contribution of each layer toward the final bitrate, as summarized in the last row.

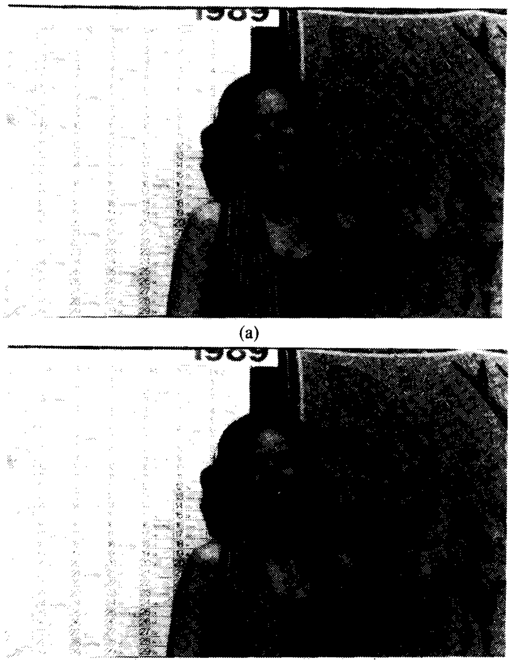
lution hierarchy remains as an interesting problem. We should note that the "optimum" must be based on a joint MR criteria: If the final layer MSE were the criteria, the optimal coder would not have an MR structure except for a restricted class of input signals. Therefore, all bits would be allocated for the final layer for most inputs. Notice that forcing higher layers to zero in a pyramid (effectively allocating no bits) makes the input signal available at the final layer (see Fig. 5).

Spatial decimation and interpolation are performed by very simple kernels similar to those of Burt and Adelson [1]. The interpolation filter involve two or three pixel averaging (recall that every other sample is zero in the upsampled signal) and is given by $[0.5 - a \ 0.5 \ 2a \ 0.5 \ 0.5 - a]$ in one-dimensional form. The parameter $0 < a < 0.5$ determines the smoothness of the interpolation kernel, and was chosen as 0.4 for the simulations. This forms a relatively smooth lowpass filter chosen so that the interpolation does not create high frequency error signals that are hard to encode with DCT.

There is also some overhead information associated with each block that has to be coded, most notably the motion vectors. They are differentially coded, with the DPCM loop initialized at the first vector at the upper left of each frame and the predictor is given by the motion vector of the previous block. The differential vectors are coded by a variable length code, typically requiring (on the average) 2 to 5 bits per vector. The horizontal and vertical displacements are coded with a 2-D variable length coder where the less likely events are coded with a prefix and PCM code. The interpolation mode is also encoded, which is one of *backward*, *forward*, or *averaged*. A runlength coder gives satisfactory results, resulting in under a bit per block overhead.

Simulations were performed for several sources. One of the sources is a non-interlaced test sequence produced at MIT that contains a rotating wheel, a subpicture from a movie, and a high detail background with artificially generated zoom and pan. A luminance-chrominance (YCrCb) version is used, with 4:2:2 chrominance subsampling. The picture size is 512 by 512, and 60 frames have been used in the simulation. Blocks of size 8×8 were used throughout, with displacement limited to ±3 at each stage. The results are summarized in Table I. The second and third columns indicate the average bits per pixel (bpp) used to encode spatial and temporal interpolation errors. Recall that each frame at the top (coarsest) layer and every other frame in the subsequent layers are spatially interpolated. Thus, the overall column is computed by averaging the two rates. The last row takes into account the overhead of coarser layers to compute the total rate in terms of bits per pixel in the finest layer.

Subjective quality of each layer was judged to be very good to excellent. No artifacts were visible at normal speed, while some DCT associated artifacts could be seen upon close examination of still frames. The average signal-to-noise (SNR) was 40.2 dB for the luminance component.



(a)

(b)

Fig. 17. Renata sequence. (a) Original frame. (b) The coded frame at 1.13 bits per pixel. (For color supplement see page 160.)

TABLE II

Layer	Spatial	Temporal	Overall
1	1.86	N/A	0.01
2	1.64	0.31	0.06
3	1.86	0.75	1.06
Total	2.07	0.76	1.13

Bit allocation to various layers for the interlaced Renata sequence. All numbers are in bits per pixel. The overall column indicates the contribution of each layer toward the final bitrate, as summarized in the last row. Note that three times as many pixels are temporally interpolated at the last layer.

The Renata (Fig. 17) sequence from RAI and Table Tennis from ISO were used for the interlaced coding simulations. A progressive subsequence was obtained by dropping odd fields, and the described algorithm was applied to code this sequence. Then a new set of motion vectors were computed to predict the odd fields from the already encoded even fields and were coded together with the prediction error. The reconstruction procedure consists of the motion-based temporal interpolation we have described. The results of a simulation using the Renata sequence are presented in Table II. The picture is 1440 by 1152 pixels, in YCrCb format with 4:2:2 chrominance subsampling. Sixteen frames have been used for the simulation. Note that unlike the non-interlaced scan case, 3/4 of the frames (or rather fields) have been temporally interpolated in the finest layer. This results in a lower overall bitrate than the previous case, with the overall column in the table reflects this weighting. The average SNR was 38.9 dB. The original picture contains a high amount of camera noise, and is therefore difficult to compress while retaining high fidelity. There were no visible artifacts in an informal subjective evaluation.

C. Relation to Emerging Video Coding Standards

The technique proposed in this paper has a number of commonalities with the emerging MPEG standard—both are based on motion compensated interpolation—and a usable subset of the techniques presented in this paper could be implemented using a MPEG like syntax. An immediate benefit would be a cost effective implementation of the algorithm with off-the-shelf-hardware.

In addition to this basic commonality, the techniques presented in this paper generalize the MPEG approach by introducing spatial hierarchies. Spatial hierarchies are particularly useful when compatibility issues require that a coded bitstream corresponding to a low resolution subsignal be easily obtainable from the higher resolution compressed video signal. They also provide a format suitable for recording on digital media, facilitating fast search and reserve playback.

Finally, this paper proposes a solution to apply the motion compensated interpolation techniques (temporal multi-resolution) to interlaced video signal. The solution fits strictly within the multi resolution, FIR approach and provides an efficient technique to deal with most signals. It is important however to continue investigating the interlaced video problem; and solutions where the two parities—odd and even field—play the same role and where the interpolation can take advantage of the nearest spatial and temporal samples available need to be investigated further.

VII. Complexity

In this section, we give a detailed analysis of the computational and memory complexity of the scheme we have presented. In particular, we compare it to conventional techniques such as full search motion estimation and hybrid DCT coding. Also of concern is the decoder simplicity. Highly asymmetric processing schemes are desirable for ATV applications. Encoders are relatively few and can have fairly complex circuitry, even non real-time algorithms can sometimes be acceptable, particularly for recording on digital media. However, decoders have to be simple, inexpensive, and must operate in real time.

Another related issue is compatibility. A low resolution decoder should not have to decode the whole bitstream at full speed to extract the low resolution information. This implies that a hierarchical coding scheme has to be employed.

A. Computational Complexity

The major computational burden is in the task of motion estimation. Before we give a complete analysis, let us first look at the major differences from a conventional predictive coding scheme. First, every other frame is interpolated temporally, so motion estimation is used half as often. Second, there are three stages that are coded in the analogous fashion, which brings the total cost to $1 + 1/8 + 1/64 = 1.14$ times the cost in the final layer.

We use a three stage search, and at each stage do a constrained search in a 7×7 window, allowing a (differential) displacement of ± 3. Thus, each stage involves 49 comparison operations. The total operation count per block is $49 + 49/4 + 49/16 = 64.31$ operations per block.

The factors are due to the fact that a block is split into 4 blocks at each coarse-to-fine step. Thus, there are four times less blocks on the second stage compared to the third (and last) stage. There is also an interpolation step at each step, but this is extremely simple, involving 2 or 4 additions per block, and can be safely neglected.

Each operation basically involves a three-step summation with appropriate shifts, as given in (9) and (8). Using 8×8 blocks, each operation thus accounts for 192 additions (assuming shifts by 2 are free). This compares with 128 additions in a conventional scheme using two frames (i.e. using the next frame is only 50% more expensive).

The maximum displacement that can be handled is $3 + 3 \cdot 2 + 3 \cdot 4 = 21$. In comparison, a full search covering the same range would require $(2 \cdot 21 + 1)^2 = 1849$ operations per block, prohibitively expensive with the current technology.

At the last step for each layer, three independent searches are performed: two conventional searches involving two frames, and one symmetric search. Thus, the number of operations is actually $49 + 49/4 + 49/16$ three-frame operations per block plus $2 \cdot 49$ two-frame operations per block.

As a final note, we should point out that the same strategy is used in coding the two coarser layers. Recalling that only half of the frames are motion-compensated, we conclude that the computational complexity of the motion estimation task is on par with the hybrid predictive type algorithms, even when hierarchical ME is used for the latter.

Once the motion vectors are computed, decoding is very simple. Interpolation mode is encoded as a side information, and all the decoder has to do is either use one displaced block from the previous or the following frame (backward or forward mode), or do a symmetrically displaced averaging (averaged mode).

B. Memory Requirement

Memory requirement of the algorithm is a critical issue from the hardware realization point of view. In what follows, we use the frame store as the basic unit, meaning memory required to hold one frame in the final layer. It is relatively easy to see that we differ from a predictive scheme in two ways (see Fig. 9):

1) Three frames are used at a time, compared to two.
2) A complete hierarchy has to be decoded, which, as we have seen, involves 15% overhead.

In the best case, no temporal interpolation is performed, so only 1.15 frames are required for the decoding. For the worst case memory usage, consider frame 1 at layer 2 (last layer) in Fig. 9. In order to decode it, following frames must also de decoded: frames 0 and 1 in layer 0; frames 0, 1 and 2 in layer 1; and frames 0, 1 and 2 in layer 2. The total number of frames stores required is $3 + 3/4 + 2/16 = 3.875$ frame stores.

In a recursive scheme, such as the hybrid DCT, only 2 frame stores are required. However, we should emphasize that the pyramid structure allows random access to any frame after decoding at most 3.875 units of input. In sharp contrast, a predictive scheme would have to decode all frames since the last restart, which might require as much as 30 frames, even if it is restarted every half second.

To conclude, overall complexity is only slightly higher than the conventional predictive schemes. In return, much faster random access is achieved, and reverse decoding is possible. Note that reverse display requires a prohibitive number of frame stores in the recursive case. The scheme is asymmetric, with a much simpler decoder, which is highly desirable in a broadcast environment. Furthermore, many of the encoding tasks can be run in parallel, making it suitable for real-time implementations.

VIII. Conclusion and Directions

We have introduced a multiresolution approach to signal representation and coding for advanced television. A three-dimensional pyramid structure has been described where motion information is used for temporal processing, in accordance with our video signal model. Very high subjective quality and SNR of over 40 dB has been achieved in coding of high resolution interlaced and non-interlaced sequences. The scheme provides many advantages in a broadcast environment or for digital storage applications:

- It is an FIR structure, and temporal processing with a short kernel ensures that no errors accumulate over time. Ability to use both the past and the future solves covered/uncovered area problems.
- Fast random access and reverse playback modes are possible.
- Different scan formats can be accommodated.
- Layered and prioritized channels ensure compatibility and graceful degradation.

In the near future at least, we are likely to have several de facto video standards. Multirate processing will be a key technique for the next generation video codecs. We have seen that the proposed scheme has a number of commonalities with the emerging MPEG standard, and can be seen as a possible evolution path. Multiresolution schemes, both conceptually and as algorithms, can be elegant solutions to a number of problems in coding and representation for advanced television.

Acknowledgment

The authors would like to thank Profs. W. Schreiber and J. Lim of MIT and Dr. M. Barbero of RAI for providing the sequences used in the simulations, and the reviewers for helpful suggestions.

References

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

[2] A. Croisier, D. Esteban, and C. Galand, "Perfect channel splitting by use of interpolation, decimation, tree decomposition techniques," *Int. Conf. on Information Sciences/Systems Rec.*, Patras, Aug. 1976, pp. 443–446.

[3] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, N.J.: Prentice-Hall, 1983.

[4] M. Vetterli, "Multi-dimensional sub-band coding: Some theory and algorithms," *Signal Processing*, vol. 6, pp. 97–112, Feb. 1984.

[5] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 1278–1288, Oct. 1986.

[6] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet decomposition," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 11, pp. 674–693, July 1989.

[7] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," submitted to *IEEE Trans. Acoust., Speech, Signal Processing*.

[8] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—a review," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.

[9] F. Glazer, G. Reynolds, and P. Anandan, "Scene matching by hierarchical correlation," in *Proc. IEEE Computer Vision and Pattern Recognition Conf.*, Washington, DC, June 1983, pp. 432–441.

[10] M. Bierling, "Displacement estimation by hierarchical blockmatching," in *Proc. SPIE Conf. on Visual Communications and Image Processing*, Boston, MA, Nov. 1988, pp. 942–951.

[11] K. Shinamura, Y. Hayashi, and F. Kishino, "Variable bitrate coding capable of compensating for packet loss," in *Proc. SPIE Conf. Visual Communications and Image Processing*, Boston, MA, Nov. 1988, pp. 991–998.

[12] M. Vetterli and K. M. Uz, "Multiresolution techniques with application to HDTV," in *Proc. Fourth Int. Colloquium on Advanced Television Systems*, Ottawa, Canada, June 1990, pp. 3B2.1–10.

[13] Motion Picture Expert Group, ISO/IEC JTC1/SC2/WG8, CCITT SGVIII, "Coded representation of picture and audio information, MPEG video simulation model two," 1990.

[14] N. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[15] M. Vetterli, "Filter banks allowing perfect reconstruction," *Signal Processing*, vol. 10, pp. 219–244, Apr. 1986.

[16] M. Vetterli and D. Anastassiou, "An all digital approach to HDTV," in *Proc. ICC-90*, Atlanta, GA, Apr. 1990, pp. 881–885.

[17] P. P. Vaidyanathan, "Quadrature mirror filter banks, m-band extensions and perfect-reconstruction technique," *IEEE ASSP Magazine*, vol. 4, pp. 4–20, July 1987.

[18] G. Karlsson and M. Vetterli, "Three dimensional sub-band coding of video," in *IEEE International Conf. on ASSP*, New York, Apr. 1988, pp. 1100–1103.

[19] W. F. Screiber and A. B. Lippman, "Single-channel HDTV systems, compatible and noncompatible," in *Signal Processing of HDTV*, L. Chiariglione, ed., Amsterdam, Netherlands: North-Holland, 1988.

[20] J. W. Woods and T. Naveen, "Subband encoding of video sequences," in *Proc. SPIE Conf. on Visual Communications and Image Processing*, Philadelphia, PA, Nov 1989, pp. 724–732.

[21] T. Kronander, "Some aspects of perception based image coding," Ph.D. thesis, Dept. of EE, Linköping University, Mar. 1989, No. 203.

[22] M. Pecot, P. J. Tourtier, and Y. Thomas, "Compatible coding of television images—Parts I and II," *Image Communication*, vol. 2, pp. 245–268, Oct. 1990.

[23] H.-M. Hang, R. Leonardi, B. G. Haskell, R. L. Schmidt, H. Bheda, and J. Othmer, "Digital HDTV compression at 44 mbps using parallel motion-compensated transform coders," in *Proc. SPIE Conf. on Visual Communications and Image Processing*, vol. 1360, Lausanne, Switzerland, Oct. 1990, pp. 1756–1772.

[24] D. Anastassiou, "Generalized three-dimensional pyramid coding for HDTV using nonlinear interpolation," in *Proc. Picture Coding Symposium*, Cambridge, MA, March 1990, pp. 1.2-1–1.2-2.

[25] K. H. Tzou, T. C. Chen, P. E. Fleischer, and M. L. Liou, "Compatible HDTV coding for broadband ISDN," in *Proc. Globecom '88*, Nov. 1988, pp. 743–749.

[26] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *IEEE International Conf. on ASSP Rec.*, Apr. 1980, pp. 291–294.

[27] D. Marr, *Vision*. San Francisco, CA: Freeman, 1982.

[28] K. M. Uz, M. Vetterli, and D. LeGall, "Multiresolution approach to motion estimation and interpolation with application to coding of digital HDTV," in *Proc. ISCAS-90*, New Orleans, LA, May 1990.

[29] P J Burt, "Multiresolution techniques for image representation, analysis, and 'smart' transmission," in *Proc. SPIE Conf. on Visual Communications and Image Processing*, Philadelphia, PA, November 1989, pp. 2–15.

[30] M. Bierling and R. Thoma, "Motion compensating field interpolation using a hierarchically structured displacement estimator," *Signal Processing*, vol. 11, pp. 387–404, Dec. 1986.

[31] K. M. Uz and D. J. LeGall, "Motion compensated interpolative coding," in *Proc. Picture Coding Symposium*, Cambridge, MA, March 1990, pp. 12.1-1–12.1-3.

[32] S. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acouts., Speech, Signal Processing*, vol. 37, pp. 2091–2110, Dec. 1989.

[33] B. Girod, "Eye movements and coding of video sequences," in *Proc. SPIE Conf. on Visual Communications and Image Processing*, Boston, MA, Nov. 1988, pp. 398–405.

[34] R. Schäfer, S. C. Chen, P. Kauff, and M. Leptin, "A comparison of HDTV-standards using subjective and objective criteria," in *Proc. Fourth Int. Colloquium on Advanced Television Systems*, Ottawa, Canada, June 1990, pp. 3B1.1-17.

[35] D. Westerkamp and H. Peters, "Comparison between progressive and interlaced scanning for a future HDTV system with digital rate reduction," in *Signal Processing of HDTV*, L. Chiariglione, Ed., Amsterdam, Netherlands: North-Holland, 1988, pp. 15–23.

imagery with application in video systems," *Proc. IEEE*, vol. 73, pp. 502-522, Apr. 1985.

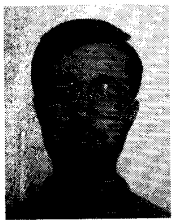
[37] CCIR, "Draft new report AD/CMTT on the digital transmission of component-coded television signals at 34 mbit/s and 45 mbit/s," CCIR Documents (1986-90) CMTT/46 and CMTT/116 + Corr 1.

[38] L. Stenger, "Digital coding of television signals—CCIR activities for standardization," *Image Communication*, vol. 1, pp. 29-43, June 1989.

[39] F.-M. Wang and D. Anastassiou, "High-quality coding of the even fields based on the odd fields of interlaced video sequences," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 140-142, Jan. 1991.

[40] Joint Photographic Expert Group, ISO/IEC JTC1/SC2/WG8, CCITT SGVIII, "JPEG technical specification, revision 5," Jan. 1990.

[41] A. Ligtenberg and G. K. Wallace, "The emerging JPEG compression standard for continuous tone images: An overview," in *Proc. Picture Coding Symposium*, Cambridge, MA, March 1990, pp. 6.1-1-6.1-4.



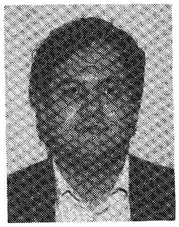
K. Metin Uz (S'88) was born in Istanbul, Turkey on October 11, 1965. He received the B.S. degrees in electrical engineering and in Physics from Boğaziçi University, Istanbul, Turkey in 1987, and the M.S. degree from Columbia University in 1988.

Since 1987, he has been employed as a Research Assistant in Center for Telecommunications Research, Columbia University, where he is working toward the Ph.D. degree. His research interests include multidimensional signal processing, multiresolution systems, and algorithms and architectures for video coding.

Martin Vetterli (S'86-M'86-SM'90) was born in Switzerland in 1957. He received the Dipl. El.-Ing. degree from the Eidgenössische Technische Hochschule Zürich, Switzerland, in 1981, the master of science degree from Stanford University, Stanford, CA, in 1982, and the Doctoral ès Science degree from the Ecole Polytechnique Fédérale de Lausanne, Switzerland, in 1986.

In 1982, he was a Research Assistant with the Computer Science Department of Stanford University, and from 1983 to 1986 he was a Researcher at the Ecole Polytechnique. He has worked for Siemens, Switzerland, and AT&T Bell Laboratories in Holmdel, NJ. Since 1986, he has been at Columbia University, New York, NY, first with the Center for Telecommunications Research and now with the Department of Electrical Engineering, where he is currently an Assistant Professor.

Dr. Vetterli is member of the editorial board of *Signal Processing* and served as European Liaison for ICASSP-88 in New York. He was recipient of the Best Paper Award of EURASIP in 1984 and of the Research Prize of the Brown Bovery Corporation (Switzerland) in 1986. His research interests include multirate signal processing, computational complexity, algorithm design for VLSI, signal processing for telecommunications, and video processing.



Didier J. LeGall (M'88) was born in Paris, France, in 1954. He received the Diplome d'Ingenieur from Ecole Centrale de Lyon, Ecully, France, in 1976, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Los Angeles (UCLA) in 1977 and 1981, respectively.

From 1982 to 1985 he was with the Medical Imaging Division of Thomson CSF, Paris, France, where he pursued research in a reconstruction and display algorithm for computerized tomography. In 1985 he joined Bell Communications Research, Morristown, NJ, first as a member of the Technical Staff working on signal processing applied to image communications, then as District Manager of the Visual Communications Group. His research interests lie in the field of signal processing, filter banks, digital image and video compression, as well as high definition television. He is also an Adjunct Professor in the Department of Electrical Engineering at Columbia University, New York.