A Common Framework for Rate and Distortion Based Scaling of Highly Scalable Compressed Video

David Taubman, *Member, IEEE*, and Avideh Zakhor, *Member, IEEE*

***Abstract*— Scalability refers to the ability to modify the resolution and/or bit rate associated with an already compressed data source in order to satisfy requirements which could not be foreseen at the time of compression. A number of researchers have already demonstrated the feasibility of efficient scalable image and video compression. The principle focus of this paper is to describe data structures for highly scalable compressed video, which are able to support simple, generic scaling approaches for both constant bit rate and constant distortion scaling criteria. Interactive video material presents particular challenges when the data stream is to be scaled to maintain an approximately constant level of distortion, rather than just a constant bit rate. Special attention is paid, therefore, to the development of generic, robust scaling algorithms for such applications. The data structures and scaling methodologies developed in this paper are particularly appealing for distribution of highly scalable compressed video over heterogeneous media, because they simultaneously support both variable bit rate (VBR) and constant bit rate (CBR) services with a wide range of available service qualities, using only simple, generic mechanisms for scaling. The performance of the proposed scaling methodologies is experimentally investigated using a highly scalable video compression algorithm, which is able to achieve comparable compression performance to that of the inherently nonscalable MPEG-1 compression standard.**

I. Introduction

In the last few years the term *scalability* has come to be associated with any of a number of desirable properties for image and, particularly, video compression algorithms. Scalability essentially means that the compressed bit stream can be manipulated in a simple manner in order to satisfy constraints on such parameters as bit rate, display resolution and frame rate, or decompression hardware complexity. In general, this manipulation consists of the extraction of relevant subsets from the compressed bit stream, each of which should represent an efficient compression of the video sequence, at some resolution and distortion. In *rate-scalability*, appropriate subsets are extracted in order to trade distortion for bit rate at some fixed display resolution. *Resolution-scalability,* on the other hand, means that subsets may be extracted which represent the video sequence at a variety of different resolutions. Rate- and resolution-scalability usually also provide a means of scaling the decompression algorithm's computational requirements. In order for the complete scalable bit stream to also represent an efficient compression of the video sequence at maximum resolution and bit rate, these subsets must be embedded within one another, rather than coexisting independently as they would in a simulcast.

The value of scalable compression lies in the fact that the bit stream may be manipulated at any point *after* the compressed bit stream has been generated. This is significant because in many important applications, advance knowledge of constraints on resolution, bit rate, or decoding complexity, may not be available during compression. Both video database servers [4] and shared digital networks can face unforeseeable throughput limitations, which jeopardize the integrity of compressed video delivered to clients. Unless the compressed data streams can be gracefully scaled down to more manageable bit rates, such potential throughput limitations can either lead to very serious corruption or else necessitate significant overallocation of resources in order to avoid the possibility of severe degradations in service quality. For these applications, rate-scalability is a highly desirable property. Scalability is also a very important property for video database, multicast, and broadcast applications with heterogeneous distribution and/or display requirements. In such heterogeneous environments, constraints on bit rate and display resolution cannot be anticipated during compression, either because the compressed data is to be stored and then retrieved under many potentially different conditions, or because the compressed data is to be simultaneously distributed to many clients, with differing display technology and/or distribution path characteristics. In these cases, both rate-scalability and resolution-scalability are desirable properties.

A number of researchers have proposed image or video compression algorithms which offer some degree of scalability. Said and Pearlman [17], Shapiro [18], and Taubman and Zakhor [24] have all proposed highly scalable algorithms for still image compression, which offer excellent compression performance over an almost continuous range of bit rate scales. With a specific view toward video applications, Bosveld *et al.* [2] and Chaddha *et al.* [3] have also proposed scalable intraframe compression schemes. Chaddha *et al.* are particularly concerned with software-only scalable compression for cooperative video applications where interframe techniques are considered too computationally expensive. Efficient, highly scalable video compression presents some additional difficulties over still image compression, because techniques based on predictive feedback for exploiting temporal redundancy do not lend themselves to scalability. This observation is

Manuscript received December 16, 1994; revised June 30, 1995. This paper was recommended by Associate Editor B. Haskell. This work was supported by Sun Microsystems, Hewlett-Packard, NSF-PYI grant MIP-9057466, California State MICRO and ONR award N00014-92-J-1732.

D. Taubman is with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA.

A. Zakhor is with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA.

Publisher Item Identifier S 1051-8215(96)04109-2.

easily understood in view of the fact that predictive coding algorithms maintain a copy of some aspect of the anticipated decoder's state, with respect to which the source signal is encoded; however, the principle behind scalability is that the decoder's state cannot be anticipated at the encoder. Nevertheless, various proposals (e.g., [6], [8], [10], [11], [28]) have been advanced for achieving limited scalability within a motion compensated predictive framework. As expected, such approaches generally suffer from rapidly escalating complexity and significant loss in compression performance as the number of available scales increases. In fact, provisions for scalable compression modes within the MPEG-2 standard, based on motion compensated prediction, explicitly restrict the number of scales to at most three [5]. For this reason, a number of researchers have proposed three-dimensional (3-D) multiresolution transforms as a vehicle for exploiting temporal redundancy without resorting to nonscalable predictive coding techniques. Among these are Sing *et al.* [19], Ohm [15], and Taubman and Zakhor [24], [25]. Such transforms are inherently much more suited to highly scalable compression than techniques based on motion compensated prediction. It should be noted, however, that highly scalable video compression also depends upon efficient layered quantization and coding strategies. In this respect, the algorithm presented in [24] is noteworthy for offering a virtually continuous range of bit rate scales. These algorithms are discussed further in Section IV.

While there is clearly much room for further investigation into highly scalable video compression algorithms, the relevance of such algorithms may in large part depend upon the ease with which scalability is able to be exploited in video storage and distribution equipment. To underscore this point, we note that the advantages of scalable compression are primarily realized by allowing such equipment to interact with the compressed video stream via scaling operations; such interaction between storage and transport entities and the traffic they support is entirely foreign to nonscalable traffic. Previous work (e.g. [11] and [15]) has focused on tailoring compression schemes to the limited native scaling potential afforded by the two priority levels offered by ATM networks. By contrast, the focus of this work is to investigate the potential of a layered substream hierarchy as an intermediate abstraction between highly scalable compression algorithms and the scaling entities associated with storage and distribution systems. While this abstraction imposes some requirements on both the compression scheme and the scaling entities, it permits simple, generic scaling operations, which are independent of syntactic features specific to any particular compression scheme. Moreover, within this abstraction, the compressed data stream may be scaled as often as desired, with either a constant bit rate or a constant level of distortion as the objective at each point. Special attention is devoted to the issues surrounding generic distortion-based scaling with hard guarantees on average bit rate properties, particularly for interactive applications.

The paper is organized as follows. In Section II we begin by outlining our proposed layered substream abstraction, together with the simple, generic scaling mechanisms supported and the requirements this abstraction imposes on the compression algorithm. Distortion based scaling is dependent on the values of *distortion tags*, which are inserted periodically into the layered substream hierarchy. Perhaps the most important question addressed by this paper is how such tags should be generated so that distortion-based scaling is able to maintain any selected measure of distortion in the reconstructed video sequence at an approximately constant level, while preserving an average bit rate interpretation that is independent of the underlying distortion measure. Section III motivates and addresses these issues. In order to place this work in a realistic context, we proceed to discuss highly scalable compression algorithms which are able to support the proposed layered substream abstraction. In particular, Section IV summarizes some of the important features of highly scalable compression schemes and briefly describes the algorithm presented in [25], which forms the context for our experimental investigations. Section V shows how the resulting compressed video data may be organized to satisfy the requirements imposed by our layered substream hierarchy. Finally, the effectiveness of rate and distortion-based scaling within the context of our layered substream abstraction are demonstrated in the experimental results of Section VI.

II. Layered Substream Hierarchy

The purpose of this section is to describe a layered substream abstraction within which simple, generic bit rate scaling may be performed according to either a constant bit rate criterion or a constant distortion criterion. Before plunging into a more thorough description of these operations, it is important to understand that rate-scalability refers to the potential to change the compressed data stream's bit rate *after* the actual compression has taken place. We refer to the actors which are able to perform this scaling as *scaling entities.* The usefulness of rate-scalability arises from the opportunity to include such scaling entities within the distribution and/or storage path of the compressed video data stream. This enables resource contention between multiple data sources to be resolved by gracefully scaling the source bit rates. Scaling entities may also be used to tailor compressed bit rates to the individual capabilities of each link in a heterogeneous multicast tree. Because we expect to include such scaling entities in the actual distribution and/or storage path of the compressed data, an important consideration is that bit rate scaling should be a generic operation, which does not depend on syntactic features specific to any particular compression algorithm. By contrast, we expect resolution-scalability and complexity-scalability to be of concern only during decompression and hence intimately dependent upon the particular compression algorithm. Thus, we are only concerned with developing a generic abstraction for rate-scalability. Given that we would like to implement rate scaling entities in the context of large public networks, a second important consideration is that these scaling entities should be as simple as possible.

Fig. 1 depicts the organization of Ψ substreams in our proposed layered hierarchy. Each substream, $\psi = 1, 2, \cdots, \Psi$, is characterized by a constant bit rate, R_ψ. In order to establish a temporal relationship between the substreams and the source video which they represent, we partition the substream

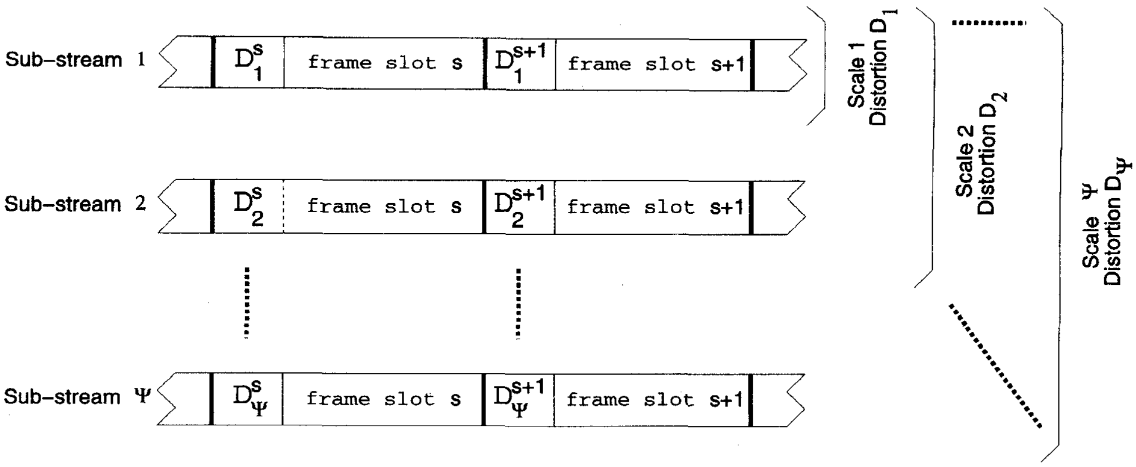


Fig. 1. Layered substream hierarchy.

hierarchy into temporal blocks, or *frame slots*, each having a duration of $\mathcal{F}$ video frame periods. The bit rates, R_ψ, are then assessed in the context of these frame slots. That is, substream ψ contains exactly $R_\psi(\mathcal{F}/F_R)$ bits in each frame slot, s, where F_R is the video frame rate. In Section I, we suggested that efficient highly scalable video compression algorithms should be based around 3-D multiresolution transforms. In fact, both memory conservation and compression efficiency considerations currently suggest that this multiresolution transform should be block-based in the temporal dimension.[1] The use of a temporally block-based transform with a block size of $\mathcal{F}'$ frames suggests a natural partitioning of the data stream into frame slots of $\mathcal{F} = K \cdot \mathcal{F}'$ frames each, where K is an integer. As we shall see, end-to-end delay is affected by the value of $\mathcal{F}$, so we select $K = 1$ for delay sensitive applications. In applications where delay is not critical, larger values of K can be helpful in enhancing the efficiency with which highly scalable compressed data may be packaged into the fixed rate substreams, ψ, within each frame slot, s.

We make two important assumptions concerning the scalable compression algorithm, whose compressed data is to be conformed to the layered hierarchy of Fig. 1. The first assumption is that the compressed data is sufficiently scalable to allow the first ψ substreams to represent an efficient compression of the original video material at rate $\sum_{\xi=1}^{\psi} R_\xi$, for each value of $\psi \in \{1, 2, \cdots, \Psi\}$. Our second assumption is that the number of substreams available for decompression may change from frame slot to frame slot, with little if any effect on the decoder's ability to utilize the entire received data stream. As we shall see, this second assumption is important in enabling effective constant distortion scaling within the same frame work as constant bit rate scaling. Suitable highly scalable compression schemes and algorithms for organizing the scalable data into substreams are discussed in Sections IV and V.

It is evident that constant bit rate scaling with a target rate of $\sum_{\xi=1}^{\psi} R_\xi$ may be accomplished simply by discarding all but the first ψ substreams in each frame slot. Scaling via substream discarding is completely generic in that it does not depend on syntactic features of the compression algorithm used to generate the substreams. With the addition of distortion tag values, $\mathcal{D}_\psi^s$, to each frame slot, s, of each substream, ψ, as shown in Fig. 1, generic distortion-based scaling is also possible within the layered substream context. The idea behind constant distortion substream scaling is that $\mathcal{D}_\psi^s$ should be representative of the distortion expected during frame slot s, when the video sequence is reconstructed from the first ψ substreams only. To obtain an approximately constant level of distortion, no greater than some distortion target, $\mathcal{D}$, it is sufficient to retain only the first $\psi^s(\mathcal{D})$ substreams of frame slot s, where

$$\psi^s(\mathcal{D}) \triangleq \min\{\psi \mid \mathcal{D}_\psi^s \leq \mathcal{D}\}. \tag{1}$$

Of course, it is unreasonable to expect truly generic scaling entities to work with complex psychovisual measures of distortion. Indeed, from a philosophical standpoint, it is probably only reasonable to expect scaling entities which reside within distribution networks to deal with quantities directly related to the bit rate. For example, a network may expect to regulate the average bit rate of a variable rate data source using some deterministic model such as a "leaky bucket" [16]. Resource allocation may also be performed on the basis of such parameters as average and peak bit rate. For these reasons, we prefer to give the distortion target, $\mathcal{D}$, a direct interpretation in terms of average bit rate. In particular, we begin by defining a standard, strictly decreasing average rate function, $R(\cdot)$, which does not depend upon any particular video sequence or compression algorithm. The significance of the average rate function, $R(\cdot)$, is that the distortion tag values, $\mathcal{D}_\psi^s$, must be selected so as to guarantee not only that video reconstructed from the first $\psi^s(\mathcal{D})$ substreams in each frame slot, s, has an approximately constant level of distortion, with respect to some reference measure of subjective distortion, but also that the average bit rate resulting from the selection of

[1] Temporally overlapping transforms not only require more memory than block based transforms, but also do not appear to offer any advantage in compression performance [15], [24].

$\psi^s(\mathcal{D})$ substreams in each frame slot, s, is equal to $R(\mathcal{D})$. That is, we first define the standard average rate function, $R(\cdot)$, and then require the distortion tag values to be chosen such that the values $\psi^s(\mathcal{D})$, $s = 1, 2, \cdots$, yielded by (1), satisfy

$$R(\mathcal{D}) = \lim_{S\to\infty} \frac{1}{S} \sum_{s=1}^{S} \left[\sum_{\xi=1}^{\psi^s(\mathcal{D})} R_\xi \right], \quad \forall \mathcal{D}. \tag{2}$$

It is important to understand that the condition, (2), need not necessarily interfere with the fact that reconstruction from the first $\psi^s(\mathcal{D})$ substreams in each frame slot, s, should provide approximately constant distortion with respect to a given reference measure. Rather, (2) states that the *level* of distortion associated with target $\mathcal{D}$ must be adjusted so as to yield an average bit rate of $R(\mathcal{D})$, regardless of the particular video sequence or reference distortion measure concerned. In this way, scaling entities may remain oblivious to the subtleties of actual measures of subjective distortion employed by the compression algorithm, provided the distortion tags consistently satisfy their average bit rate interpretation, as given by (2).

In practice, of course, the distortion target, $\mathcal{D}$, and distortion tag values, $\mathcal{D}^s_\psi$, may only assume values from a discrete set. In the remainder of this paper, therefore, we adopt the following notation. The distortion target, $\mathcal{D}$, takes on one of p values, $d_1 > d_2 > \cdots > d_p$, satisfying $R(d_1) \geq R_1$ and $R(d_p) \leq R_\Psi$.[2] In view of (1), it is sufficient to consider distortion tag values belonging to the augmented set, $\mathcal{D}^s_\psi \in \{d_0, d_1, \cdots, d_p\}$, where d_0 is any value which exceeds the largest distortion target, i.e., $d_0 > d_1$. To see why this is a sufficient set of distortion tag values, observe that any tag value, $\mathcal{D}^s_\psi$ in the interval $(d_{i+1}, d_i]$ may be replaced with the value $\mathcal{D}^s_\psi = d_i$, without interfering with the result of constant distortion scaling with any target, $\mathcal{D} \in \{d_1, d_2, \cdots, d_p\}$. Note that, with the largest possible distortion target, $\mathcal{D} = d_1$, constant distortion scaling discards all substreams after the first substream found to have a distortion tag value less than d_0. A distortion tag value of $\mathcal{D}^s_\psi = d_0$ may thus be interpreted as indicating that ψ substreams are not sufficient to satisfy any of the valid distortion targets $d_1, d_2, \cdots, d_p$ in frame slot s. If $R(d_1) > R_1$, some distortion tags must assume the value d_0 from time to time. On the other hand, if $R(d_1) = R_1$, so that the minimum average bit rate and the minimum instantaneous bit rate are identical, we can have $\psi^s(d_1) = 1, \forall s$, which means that the distortion tag value d_0 need never be used.

Although the distortion-based scaling algorithm embodied by (1) is necessarily very simple and its average rate interpretation, given in (2), is independent of any particular video sequence or reference measure of actual subjective distortion, the mechanisms used to generate appropriate distortion tags, $\mathcal{D}^s_\psi$, need not be. Techniques for generating meaningful distortion tag values for both interactive and prerecorded video applications are discussed next in Section III.

Before concluding this section, we briefly consider the implications of our proposed layered substream hierarchy for end-to-end delay. Here we are concerned only with *inherent delay*, by which we mean the minimum end-to-end delay achievable, assuming that physical transmission delays and computation times are negligible. Without any loss of generality, we may assume that video frames $s\mathcal{F}$ through $(s+1)\mathcal{F}-1$ can be reconstructed and displayed if and only if the compressed data in frame slot s is available at the receiver.[3] Moreover, because we do not impose any constraints on the proportion of bits devoted to representing each of these frames, we cannot guarantee that any of them will be available for display until the entire frame slot has been received. Similarly, the compression algorithm might have to wait until the arrival of all source video on which the compressed data for frame slot s depends before it decides how best to allocate the $R_\psi\,(\mathcal{F}/F_R)$ bits in each substream, ψ.[4] This means that none of frame slot s can be generated until at least the arrival of source frame $(s+1)\mathcal{F} - 1$. If the compression algorithm employs an overlapping temporal transform, then even later source frames may be required before the frame slot can be generated. As noted at the beginning of this section, however, there is little reason to use anything other than a temporally block-based transform. In this case, by setting $\mathcal{F}$ equal to an integral multiple of the temporal block size, we can ensure that source frame $(s+1)\mathcal{F} - 1$ is always the last frame in a transform block so that no further delay is introduced by the multiresolution transform. In summary, none of frame slot s can be generated until source frame $(s+1)\mathcal{F} - 1$ arrives, after which we must wait $\mathcal{F}$ frame periods for the fixed rate substreams of frame slot s to be completely transmitted to the receiver; only then do we have any guarantee that any of frames $s\mathcal{F}$ through $(s+1)\mathcal{F} - 1$ can be decompressed and displayed. Consequently, frame $s\mathcal{F}$ experiences the maximum end-to-end delay of $2\mathcal{F} - 1$ frame periods.

III. Generation of Distortion Tags

In Section II, we showed how constant distortion scaling may be performed in the context of our proposed layered substream abstraction. We turn our attention now to the task of generating the distortion tags, $\mathcal{D}^s_\psi \in \{d_0, d_1, \cdots, d_p\}$, which determine the behavior of this scaling operation. We assume that the compression algorithm is able to assign a reference distortion value, V^s_ψ, to each substream, ψ, in each frame slot, s. V^s_ψ is some measure of the average distortion expected over frame slot s, when the video sequence is reconstructed from substreams $1, 2, \cdots, \psi$. We refer to the reference distortion measure used to generate these V^s_ψ values as the V-distortion measure. V-distortion measures may vary from a simple mean squared error (MSE) estimate to more complex perceptually based distortion measures. The idea is to determine a strictly increasing map, $\mathcal{T}$, from reference distortion values, V^s_ψ, to distortion tag values, $\mathcal{D}^s_\psi \in \{d_0, d_1, \cdots, d_p\}$, such that constant distortion scaling is guaranteed to satisfy the average bit

[2] The minimum average bit rate, $R(d_1)$, must clearly be no smaller than the minimum instantaneous bit rate, R_1. Similarly, the largest average bit rate, $R(d_p)$, may not exceed the largest instantaneous bit rate, R_Ψ.

[3] The validity of this statement depends only on selecting an appropriate point at which to start numbering the frames and observing that the frame slots are separated by exactly $\mathcal{F}$ frame periods.

[4] In our experience, premature allocation of the number of bits used to represent different frames can significantly degrade the overall efficiency of the compressed video representation.

rate requirement of (2). In this way, constant distortion scaling holds the distortion of the reconstructed video approximately constant with respect to the particular V-distortion measure selected. That is, $V^s_{\psi^s(\mathcal{D})}$ should be approximately constant from frame slot to frame slot, for any distortion target $\mathcal{D} \in \{d_1, d_2, \cdots, d_p\}$.

We begin, in Section III-A, by considering the determination of this map, $\mathcal{T}$, when the reference distortion values, V^s_ψ, are known ahead of time for all substreams, ψ, and frame slots, s. Of course, such an approach is only applicable for prerecorded video material of finite duration. For interactive applications, distortion tag values, $\mathcal{D}^s_\psi$, in frame slot s must be determined without any information about the reference distortion values in future frame slots, $s+1$, $s+2, \cdots$. Section III-B discusses an adaptive strategy for such applications, in which an adaptive map $\mathcal{T}^s$, from reference distortion values, V^s_ψ, to distortion tag values, $\mathcal{D}^s_\psi$, is allowed to change slowly from frame slot to frame slot. In this case, the rate at which $\mathcal{T}^s$ is allowed to change determines the time frame over which distortion can be considered to be held constant by constant distortion scaling.

A. Generation of Distortion Tag Values for Prerecorded Video

In this section, we discuss the determination of the strictly increasing map, $\mathcal{T}$, from reference distortion values, V^s_ψ, to distortion tag values, $\mathcal{D}^s_\psi$, in the case of a prerecorded video sequence consisting of exactly S frame slots. Because the video sequence is prerecorded, the set of all reference distortion values $\{V^s_\psi \,|\, 1 \leq s \leq S, 1 \leq \psi \leq \Psi\}$ may be employed to construct this map, $\mathcal{T}$. Our objective is to select the map, $\mathcal{T}$, for which the average bit rate over all S frame slots

$$\frac{1}{S}\sum_{s=1}^{S}\left[\sum_{\xi=1}^{\psi^s(d_i)} R_\xi\right] \tag{3}$$

is as close as possible to, but no larger than $R(d_i)$, for each distortion target $d_1, d_2, \cdots, d_p$. Recall that $R(\cdot)$ is our standard average rate function, which does not depend upon the video sequence or the reference V-distortion measure, whereas the map, $\mathcal{T}$, depends on both the video sequence and the V-distortion measure, through the V^s_ψ values. Because only a finite number of frame slots are available, and substream discarding allows for only a discrete set of bit rates in any frame slot, it is not generally possible to obtain exact equality between the short term average bit rate of (3) and the nominal average bit rate, $R(\mathcal{D})$. As the number of frame slots, S, becomes increasingly large, however, the discrepancy between $R(\mathcal{D})$ and the average in (3) rapidly becomes negligible so that (2) holds.

We observe that $\mathcal{T}$ is simply a quantization operator, quantizing continuous V-distortion values onto the discrete set of distortion tag values, $\{d_0, d_1, \cdots, d_p\}$. As such, we may characterize $\mathcal{T}^s$ by p thresholds, $t_1 > t_2 > \cdots > t_p$, according to

$$\mathcal{T}^{-1}(d_i) = (t_{i+1}, t_i], \quad i = 0, 1, 2, \cdots, p, \tag{4}$$

where we have used $t_0 = \infty$ and $t_{p+1} = -\infty$, for notational convenience. Thus, $\mathcal{T}(v) = d_i$ whenever the reference distortion v is less than or equal to the threshold t_i but greater than t_{i+1}. The following observation demonstrates the usefulness of characterizing $\mathcal{T}$ by (4).

Observation 1: The number of substreams $\psi^s(d_i)$ retained during constant distortion scaling in frame slot s, with distortion target $\mathcal{D} = d_i$, depends only upon the threshold, t_i, according to

$$\psi^s(d_i) = \min\{\psi \,|\, V^s_\psi \leq t_i\}, \quad i = 1, 2, \cdots, p. \tag{5}$$

The proof of this statement may be found in the Appendix.

This observation is particularly helpful because it indicates that the values, $\psi^s(d_i)$, and hence the average bit rate of (3), depend only upon the threshold value t_i and not on the values of t_j, $j \neq i$. In order to guarantee that this average rate is as close as possible to $R(d_i)$, without exceeding $R(d_i)$, we have simply to select

$$t_i = \min\left\{t \,\middle|\, \sum_{s=1}^{S}\left(\sum_{\xi=1}^{\min\{\psi|V^s_\psi\leq t\}} R_\xi\right) \leq SR(d_i)\right\}, \quad i = 1, 2, \cdots, p. \tag{6}$$

Equation (6) is easily understood by observing that the average bit rate associated with distortion target d_i is a nonincreasing function of t_i. Therefore, we wish to select the smallest possible distortion threshold, t_i, such that the average bit rate does not exceed $R(d_i)$. Evaluation of the threshold values t_i according to (6), grows rapidly in complexity as S becomes large. While this computation is found to be quite manageable for the relatively short video sequences investigated in Section VI-D, the adaptive approach described in Section III-B is probably more suitable for very long video sequences, whether they are prerecorded or not. Nevertheless, the above algorithm serves as a useful introduction to the less obvious algorithm described in Section III-B.

B. Generation of Distortion Tag Values for Interactive Applications

In Section III-A, we considered the determination of a single map, $\mathcal{T}$, from reference distortion values, V^s_ψ, to distortion tag values, $\mathcal{D}^s_\psi$, so as to satisfy the average bit rate requirement of (2) for each distortion target, $\mathcal{D} \in \{d_1, d_2, \cdots, d_p\}$. The map, $\mathcal{T}$, necessarily depends upon the set of all reference distortion values, V^s_ψ, over all substreams and all frame slots in the video sequence. Such information is not available in interactive applications. Thus, it is necessary to consider an adaptive map, $\mathcal{T}^s$, such that $\mathcal{D}^s_\psi = \mathcal{T}^s(V^s_\psi)$, $\forall\, \psi$, s and $\mathcal{T}^s$ is allowed to change from frame slot to frame slot. Because the map is not fixed ahead of time, we are able to guarantee that (2) is satisfied. On the other hand, because the map is not fixed, the time period, over which substream scaling according to (1) can be considered to hold V-distortion approximately constant, depends upon the rate at which $\mathcal{T}^s$ is allowed to change. These concepts shall be made more concrete as we describe our proposed approach to the adaptation of $\mathcal{T}^s$.

Before discussing the adaptation of $\mathcal{T}^s$, we observe that the average rate constraint of (2) offers no indication as to the time over which it may be enforced. The sequences considered in

Section III-A have a known finite duration of S frame-slots, for which the averaging period is made explicit in (3). For interactive applications, however, the duration of the video sequence is generally unknown. Moreover, even for prerecorded video material, it may be necessary to allow average bit rate properties to be verified, e.g., by network regulatory entities, within a shorter time frame than the duration of the entire video sequence. For these reasons, we impose a tighter requirement on the average bit rate interpretation of the distortion targets $\mathcal{D} \in \{d_1, d_2, \cdots, d_p\}$. In particular, we require

$$\left\| \left[\sum_{s=1}^{S} \sum_{\psi=1}^{\psi^s(d_i)} R_\psi \right] - SR(d_i) \right\| \cdot \frac{\mathcal{F}}{F_R} \le R(d_i)B, \qquad i = 1, 2, \cdots, p, \quad \forall S \ge 1 \tag{7}$$

where B is a fixed parameter, whose interpretation will become apparent presently. By inspection of (7), B must have the dimension of time, measured in seconds.

Dividing both sides of (7) by S and taking the limit as $S \to \infty$, it is clear that (7) implies (2). In order to appreciate the significance of (7), note that $\mathcal{F}/F_R$ is the duration of each frame slot in seconds. Thus, the left hand side of (7) is equal to the difference between the number of bits required by the first S frame slots of the scaled data stream, with distortion target d_i, and the number of bits which would be required if the data stream had a constant bit rate of $R(d_i)$. As such, (7) may be recognized as a leaky bucket condition [16]. In particular, (7) states that a leaky bucket, which is initially filled to half of a total capacity of $2R(d_i)B$ bits, and which leaks at a constant rate of $R(d_i)$ b/s, should neither underflow nor overflow as it is filled with the constant distortion scaled data stream with distortion target, $\mathcal{D} = d_i$. We note that leaky bucket models have been proposed for average bit rate regulation in shared networking environments [16]. The $R(d_i)$ term on the right hand side of (7) ensures that the bucket capacity is proportional to the average bit rate. The constant parameter, B, may thus be understood as an indication of the time frame, over which the average bit rate interpretation of any distortion target may be enforced, say by network policing entities. For example, a data stream, whose instantaneous bit rate continually exceeds its nominal average bit rate by 100% may be detected as violating the condition of (7) after B seconds. If it continually exceeds its average rate by only 50%, policing entities may identify it as a delinquent data source only after $2B$ seconds have elapsed. More generally, (7) states that the average bit rate, taken over the first T seconds, must be within $B/T \times 100\%$ of the nominal average bit rate, $R(d_i)$.

We are now in a position to discuss adaptation of the map, $\mathcal{T}^s$. In order to appreciate our proposed approach, it is helpful to understand the nature of this adaptation problem. Exactly as in Section III-A, $\mathcal{T}^s$ is a quantization operator, which may be characterized by the p thresholds, $t_1^s > t_2^s > \cdots > t_p^s$, such that

$$(\mathcal{T}^s)^{-1}(d_i) = (t_{i+1}^s, t_i^s], \quad i = 0, 1, 2, \cdots, p.$$

Thus, we are faced, in general, with the problem of jointly adapting these p thresholds, in order to satisfy the p constraints of (7).[5] Moreover, in order to keep distortion as constant as possible, it is important that the map be adapted as slowly as possible without violating any of these p constraints. Thus, a truly optimal adaptation scheme would be expected to satisfy all p constraints tightly.[6] Referring to Observation 1, we see that each sequence, $\psi^s(d_i)$, $s = 1, 2, \cdots$, depends only upon the corresponding sequence of threshold values, t_i^s, $s = 1, 2, \cdots$. Thus, each of the p constraints of (7) is independently controlled by one of the adaptive thresholds, t_i^s. This suggests that an optimal adaptation scheme should independently adapt each threshold, t_i^s, as slowly as possible without violating the corresponding constraint in (7), thereby ensuring that each of the constraints in (7) is tight. Although this reasoning may appear to significantly simplify the adaptation task, it is important to bear in mind that the p thresholds are in fact coupled by the $p-1$ ordering constraints, $t_1^s > t_2^s > \cdots > t_p^s$. Only in this context does Observation 1 hold. As it turns out, the need to avoid misordering of the threshold values considerably complicates the map adaptation task, motivating the somewhat indirect approach proposed in the remainder of this section.

It is convenient to represent the adaptive quantizing map, $\mathcal{T}^s$, indirectly as the composition of a fixed, continuous map, $\mathcal{M}$, followed by an adaptive quantization operator, $\mathcal{A}^s$, i.e., $\mathcal{T}^s = \mathcal{A}^s \circ \mathcal{M}$. As we shall see, this approach allows us to describe a simple, stable scheme for independently adapting the quantization thresholds of $\mathcal{A}^s$, so as to avoid misordering difficulties, while the implications of this adaptive scheme for bit rate and distortion properties may be controlled by appropriate selection of the fixed part, $\mathcal{M}$. That is, the $p-1$ threshold ordering constraints are satisfied by appropriate adaptation of $\mathcal{A}^s$, whereas we arrange for the p leaky bucket constraints of (7) to be satisfied tightly by appropriate design of $\mathcal{M}$.

In our formulation, $\mathcal{M}$ is a strictly decreasing, continuous map; we write $\mathcal{M}(V_\psi^s) = \Phi_\psi^s, \forall \psi, s$, where Φ_ψ^s may be thought of as a measure of the *fidelity* associated with video reconstructed from the first ψ substreams in frame slot s. The adaptive part, $\mathcal{A}^s$, is then a quantization operator, mapping the continuous valued fidelity values, Φ_ψ^s, onto the discrete set of distortion tag values, $\{d_0, d_1, \cdots, d_p\}$, i.e., $\mathcal{D}_\psi^s = \mathcal{A}^s(\Phi_\psi^s)$. We choose $\mathcal{M}$ to be a decreasing map because the ensuing arguments are more intuitive when the intermediate variables Φ_ψ^s may be interpreted as fidelity values; specifically, fidelity increases with substream number, ψ. Strictly by way of example, the reference distortion values, V_ψ^s, might represent MSE, whereas the fidelity values, Φ_ψ^s, might represent peak signal-to-noise ratio (PSNR), in which case the strictly decreasing map, $\mathcal{M}$, would be given by $\mathcal{M}(v) = 10 \log_{10}(255^2/v)$. Because $\mathcal{M}$ is strictly decreasing and continuous, it must be invertible. Thus, the task of holding V-distortion constant is identical to that of holding fidelity constant.

[5] There is one constraint for each of the p distortion targets, $\mathcal{D} = d_1, d_2, \cdots, d_p$.

[6] If this were not so, then it should be possible to hold distortion more constant for one or more of the distortion targets by allowing greater variations in the short term average bit rate reflected by the left hand side of (7).

We divert our attention now to the adaptive quantization operator, $\mathcal{A}^s$. $\mathcal{A}^s$ may be characterized by the p fidelity thresholds, $a_1^s < a_2^s < \cdots < a_p^s$, according to

$$(\mathcal{A}^s)^{-1}(d_i) = [a_i^s, a_{i+1}^s), \quad i = 0, 1, \cdots, p \tag{8}$$

where we have used $a_0^s = -\infty$ and $a_{p+1}^s = \infty$ for notational convenience. Notice that the fidelity thresholds, a_i^s, are related to the distortion thresholds, t_i^s, of the composite map, $\mathcal{T}^s$, according to $t_i^s = \mathcal{M}(a_i^s)$. The following observation demonstrates the usefulness of this characterization of $\mathcal{A}^s$ in terms of the thresholds, a_i^s.

Observation 2: The number of substreams, $\psi^s(d_i)$, retained during constant distortion scaling in frame slot s, with distortion target $\mathcal{D} = d_i$, depends only upon the threshold, a_i^s, according to

$$\psi^s(d_i) = \min\{\psi \mid \Phi_\psi^s \geq a_i^s\}, \quad i = 1, 2, \cdots, p. \tag{9}$$

The proof is essentially identical to that of Observation 1, found in the Appendix.

Observation 2 is important because it indicates that the validity of the leaky bucket model of (7), for any distortion target, $\mathcal{D} = d_i$, depends only upon the corresponding sequence of threshold values, a_i^s, $s = 1, 2, \cdots$, i.e., it does not depend upon the other threshold values, a_j^s, $j \neq i$. This means that our adaptation problem for the map $\mathcal{A}^s$ may be considered as p independent adaptation problems: for distortion target d_i, we update the fidelity threshold, a_i^s, from frame slot to frame slot, according to the value of $\psi^s(d_i)$, in such a way as to guarantee that (7) is satisfied. The proposed adaptation scheme itself is presented in Section III-B-1. It is important to remember that these p adaptation tasks are coupled by the $p-1$ ordering constraints, $a_1^s < a_2^s < \cdots < a_p^s$. If any of these ordering constraints are violated at any point, then (9) becomes invalid. Section III-B-2 takes up this issue, demonstrating that the adaptation scheme of Section III-B-1 preserves this ordering except under rare circumstances. In these rare cases, a slight perturbation in the reference distortion values V_ψ^s is sufficient to guarantee that the ordering of the fidelity threshold values is never violated. Moreover, the probability that such corrective distortion perturbations need be applied, as well as the magnitude of the perturbations, both decrease rapidly to zero, as the time constant, B, of (7), increases.

1) Proposed Adaptation Scheme: Our proposed adaptation strategy is based on the observation that increasing the value of a_i^s results in an increase in the average fidelity corresponding to distortion tag d_i, which tends to increase the average bit rate associated with the distortion target, d_i. Thus, whenever the instantaneous bit rate associated with a distortion target d_i is found to be lower than $R(d_i)$, we set a_i^{s+1} to be a little larger than a_i^s; whenever it is found to be larger than $R(d_i)$, we set a_i^{s+1} to be a little smaller than a_i^s. To be precise, we simply update each of the parameters a_i^s according to

$$a_i^{s+1} = a_i^s - \left[\sum_{\psi=1}^{\psi^s(d_i)} R_\psi - R(d_i)\right], \quad i = 1, 2, \cdots, p. \tag{10}$$

In this case, the amount by which the ith fidelity threshold changes between frame slots s and $s+1$ is exactly equal to the difference between the average and instantaneous bit rates associated with distortion target d_i in frame slot s. We point out that the relative impact of this change on the threshold value may be made as small as desired by appropriate choice of the map, $\mathcal{M}$. This is because the definition of fidelity is controlled by $\mathcal{M}$. Scaling $\mathcal{M}$ causes all fidelity values to be correspondingly scaled. Thus, the rate at which the composite map $\mathcal{T}^s$ adapts in response to (10) depends upon the choice of $\mathcal{M}$. Our next task is to show how $\mathcal{M}$ should be selected in order to satisfy each of the p leaky bucket constraints in (7) tightly.

Suppose the fidelity values Φ_ψ^s associated with each substream, ψ, are bounded according to $\Phi_\psi^{\min} \leq \Phi_\psi^s \leq \Phi_\psi^{\max}$, $\forall\, s$. We show now that these bounds play a key role in determining whether or not (7) is satisfied. Moreover, because $\Phi_\psi^{\min}$ and $\Phi_\psi^{\max}$ depend upon the map $\mathcal{M}$, we shall ultimately be able to satisfy (7) by appropriate choice of $\mathcal{M}$. We begin with the following observation.

Observation 3: The fidelity threshold value, a_i^s, adapted according to (10) is bounded according to $a_i^{\min} \leq a_i^s \leq a_i^{\max}$, where $a_i^{\min}$ and $a_i^{\max}$ satisfy

$$a_i^{\min} > \Phi^{\min}_{\psi_i^b} + \left[R(d_i) - \sum_{\xi=1}^{\Psi} R_\xi\right] \tag{11}$$

with

$$\psi_i^b = \max\left\{\psi \,\middle|\, \sum_{\xi=1}^{\psi} R_\xi \leq R(d_i)\right\}$$

and

$$a_i^{\max} < \Phi^{\max}_{\psi_i^a} + [R(d_i) - R_1] \tag{12}$$

with

$$\psi_i^a = \min\left\{\psi \,\middle|\, \sum_{\xi=1}^{\psi} R_\xi \geq R(d_i)\right\}$$

provided the initial value, a_i^1, is selected to be anywhere within these bounds. In (11) and (12), the constants ψ_i^b and ψ_i^a represent the number of substreams corresponding to the maximum instantaneous bit rate not exceeding $R(d_i)$ and the minimum instantaneous bit rate not less than $R(d_i)$, respectively. The superscripts, b and a, are intended to suggest the adjectives *below* and *above*, respectively. The proof of this observation may be found in the Appendix.

In view of (10), the terms $[R(d_i) - R_1]$ and $[\sum_{\xi=1}^{\Psi} R_\xi - R(d_i)]$ in (12) and (11) represent, respectively, the maximum possible increase and decrease of the threshold, a_i^s, between two consecutive frame slots. The magnitude of these terms should, in practice, be very much smaller than the a_i^s values themselves, because constant distortion scaling is only effective if the map, $\mathcal{A}^s$, used to generate the distortion tag values, changes slowly from frame slot to frame slot. As a result, the bounds, $a_i^{\max}$ and $a_i^{\min}$, should usually differ only negligibly from the values of $\Phi^{\max}_{\psi_i^a}$ and $\Phi^{\min}_{\psi_i^b}$, respectively.

Observation 3 allows us to transform (7) into a requirement on the bounds $\Phi_\psi^{\max}$ and $\Phi_\psi^{\min}$, using the adaptation scheme of (10), as follows:

$$\begin{aligned}
&\left\| \sum_{s=1}^{S} \sum_{\psi=1}^{\psi^s(d_i)} R_\psi - SR(d_i) \right\| \\
&\quad = \left\| \sum_{s=1}^{S} [R(d_i) + a_i^s - a_i^{s+1}] - SR(d_i) \right\| \\
&\quad = \|a_i^{S+1} - a_i^1\| \\
&\quad \le a_i^{\max} - a_i^{\min} < (\Phi_{\psi_i^a}^{\max} - \Phi_{\psi_i^b}^{\min}) + \sum_{\xi=2}^{\Psi} R_\xi, \\
&\quad \forall S, \quad 1 \le i \le p. \qquad (13)
\end{aligned}$$

Note that, in accordance with the discussion above, we expect the right hand side of (13) to be dominated by the term $(\Phi_{\psi_i^a}^{\max} - \Phi_{\psi_i^b}^{\min})$ in practical applications. Equation (13) indicates that the p leaky bucket constraints of (7) are satisfied if and only if

$$\Phi_{\psi_i^a}^{\max} - \Phi_{\psi_i^b}^{\min} \le \left(B \frac{F_R}{\mathcal{F}} \right) R(d_i) - \sum_{\xi=2}^{\Psi} R_\xi, \quad i = 1, 2, \cdots, p. \qquad (14)$$

Thus, each of the leaky bucket constraints is satisfied tightly, if and only if equality holds in (14). Again, for practical applications, we note that B should usually be sufficiently large that the term $\sum_{\xi=2}^{\Psi} R_\xi$ on the right hand side of (14) is insignificant.

Equation (14) suggests that the leaky bucket constraints of (7) might be satisfied by appropriate design of the strictly decreasing function $\mathcal{M}$, which maps reference distortion values into fidelity values. The idea is to first obtain bounds, $V_\psi^{\min}$ and $V_\psi^{\max}$, for the V distortion associated with the first ψ substreams of the layered hierarchy. In interactive applications it is not usually possible to predict the exact range of reference distortion values that may arise; however, experience may be used to identify and then enforce appropriate bounds. For example, we might measure the V_ψ^s values over some representative collection of video sequences and then select $V_\psi^{\min}$ and $V_\psi^{\max}$ such that $V_\psi^s \ge V_\psi^{\min}$ for 99% of all frame slots s, and $V_\psi^s \le V_\psi^{\max}$ for 99% of all frame slots. Having selected $V_\psi^{\min}$ and $V_\psi^{\max}$, using this or some other method, we simply hard-limit the V_ψ^s values determined during compression so as to guarantee that $V_\psi^{\min} \le V_\psi^s \le V_\psi^{\max}$ without exception. Thus, the reference distortion values must occasionally be artificially constrained to lie within the selected bounds; however, if the training video sequences used to determine $V_\psi^{\min}$ and $V_\psi^{\max}$ are truly representative, this hard-limiting need not significantly interfere with the video sequence distortion associated with constant distortion scaling. Having fixed these reference distortion bounds, we have simply to choose a continuous, strictly decreasing map, $\mathcal{M}$, such that

$$\mathcal{M}(V_{\psi_i^a}^{\min}) - \mathcal{M}(V_{\psi_i^b}^{\max}) \le \left(B \frac{F_R}{\mathcal{F}} \right) R(d_i) - \sum_{\xi=2}^{\Psi} R_\xi, \quad i = 1, 2, \cdots, p. \qquad (15)$$

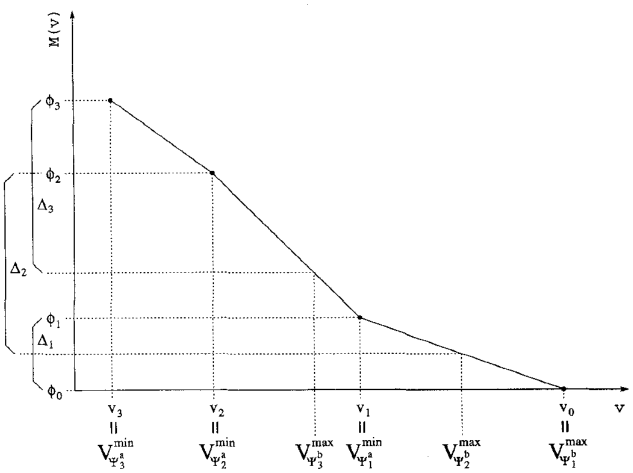


Fig. 2. Piecewise linear, decreasing map, $\mathcal{M}$, from reference distortion values, V_ψ^s, to fidelity values, Φ_ψ^s.

Recalling that we would like to satisfy all p leaky bucket constraints tightly, so as to avoid overly constraining the instantaneous bit rate associated with any distortion target, our objective is to select $\mathcal{M}$ such that equality holds in (15). Intuitively, there should be enough degrees of freedom to make such a selection, provided the indices ψ_i^a are all distinct, i.e., $i \ne j \implies \psi_i^a \ne \psi_j^a$. The specific choices described in Section VI-A certainly have this property. In order to make these concepts concrete, we now describe a simple, piecewise linear map, $\mathcal{M}$, to guarantee equality in each of the p constraints of (15).

Consider a piecewise linear map $\mathcal{M}$, mapping intervals $[v_i, v_{i-1}]$ linearly onto the intervals $[\phi_{i-1}, \phi_i]$, $i = 1, 2, \cdots, p$, as shown in Fig. 2 for $p = 3$. The constants v_i and ϕ_i may be obtained using the following constructive algorithm. Fig. 2 provides a useful framework in which to appreciate the operation of this simple algorithm.

- Set $v_0 = V_{\psi_1^b}^{\max}$ and $\phi_0 = 0$. That is, $\mathcal{M}(V_{\psi_1^b}^{\max}) = 0$.
- For each $i = 1, 2, \cdots, p$
  - — Evaluate $\Phi_{\psi_i^b}^{\min} = \mathcal{M}(V_{\psi_i^b}^{\max})$ from that part of the piecewise linear map constructed so far. Then set $v_i = V_{\psi_i^a}^{\min}$ and $\phi_i = \mathcal{M}(V_{\psi_i^b}^{\max}) + \Delta_i$, where $\Delta_i \triangleq [B(F_R/\mathcal{F})] R(d_i) - \sum_{\xi=2}^{\Psi} R_\xi$. This guarantees that equality holds in (15) for distortion target d_i.

Notice that the above algorithm relies upon the fact that the indices ψ_i^a are distinct, so that the critical points, v_i, can also be distinct.

We may now interpret the parameter B in terms of the rate at which the thresholds a_i^s are allowed to change.[7] In particular, suppose the instantaneous bit rate $\psi^s(d_i)$, associated with distortion target $\mathcal{D} = d_i$, exceeds the nominal average bit rate, $R(d_i)$, by 100% in some frame slot s. Then a_i^s decreases by an amount $R(d_i)$ between frame slots s and $s + 1$. On the

[7] Recall that B can also be interpreted as the number of seconds required to detect the fact that a data stream, whose instantaneous bit rate continually exceeds the nominal average rate by 100%, is in violation of the leaky bucket criterion of (7).

other hand, from (11), (12), and (15), the maximum range of a_i^s values is $\overset{<}{\approx} (\Phi^{\max}_{\psi_i^a} - \Phi^{\min}_{\psi_i^b} + \sum_{\xi=2}^{\Psi} R_\xi) = BR(d_i)\,(F_R/\mathcal{F})$. So, in the interval from frame slot s to frame slot $s+1$, a_i^s changes by approximately $(F/F_R)\,(1/B)$ times its maximum range; this frame slot interval has a duration of F/F_R seconds. More generally, if the instantaneous bit rate associated with distortion target $\mathcal{D} = d_i$ exceeds $R(d_i)$ by 100% for T seconds, the corresponding change in a_i^s is T/B times its maximum range. In this way, B may be interpreted as an adaptation time constant. Large values of B allow the map $\mathcal{A}^s$, and hence $\mathcal{T}^s$ to adapt more slowly, so that the V-distortion of reconstructed video is held approximately constant over a longer period of time. On the other hand, large values of B imply that policing agents will require more time to detect delinquent data sources in a shared networking environment.

2) Threshold Ordering and Other Considerations: In Section III-B-1, we proposed an algorithm for adaptively updating the map $\mathcal{A}^s$ in each frame slot s and selecting an appropriate fixed map, $\mathcal{M}$, so as to satisfy (7). In developing this algorithm, we assumed that the adaptation strategy of (10) preserves the order $a_1^s < a_2^s < \cdots < a_p^s$ of the fidelity threshold values. This assumption is critical to the arguments above, because the fidelity threshold bounds of Observation 3 depend upon the validity of (9) which, in turn, depends upon the fact that $a_1^s < a_2^s < \cdots < a_p^s$. The purpose of this section is to examine the validity of this assumption and show how it may be enforced.

We may assume that the initial threshold values are selected such that $a_1^1 < a_2^1 < \cdots < a_p^1$. It is sufficient, therefore, to ensure that $(a_{i+1}^{s+1} - a_i^{s+1}) > 0$ whenever $(a_{i+1}^s - a_i^s) > 0$ for $i = 1, 2, \cdots, p-1$. To this end, suppose that $(a_{i+1}^s - a_i^s) > 0$, for $1 \le i < p$ in some frame slot s, and observe from (10) that

$$\begin{aligned}(a_{i+1}^{s+1} - a_i^{s+1}) &= (a_{i+1}^s - a_i^s) + [R(d_{i+1}) - R(d_i)] \\ &\quad - \sum_{\xi=\psi^s(d_i)+1}^{\psi^s(d_{i+1})} R_\xi. \qquad (16)\end{aligned}$$

Moreover, because (9) holds during frame slot s, the event $\psi^s(d_i) < \psi^s(d_{i+1})$ can occur only if $a_i^s \le \Phi^s_{\psi^s(d_i)} < a_{i+1}^s$.[8] Thus, the term $\sum_{\xi=\psi^s(d_i)+1}^{\psi^s(d_{i+1})} R_\xi$ in (16) is zero unless $a_i^s \le \Phi_\psi^s < a_{i+1}^s$ for some substream ψ. Noting that $[R(d_{i+1}) - R(d_i)] > 0$ and $(a_{i+1}^s - a_i^s) > 0$, (16) indicates that the event, $(a_{i+1}^{s+1} - a_i^{s+1}) \le 0$ cannot occur unless $(a_{i+1}^s - a_i^s)$ is very small already[9] and at least one substream, ψ, has its fidelity value, $\Phi_\psi^s \in [a_i^s, a_{i+1}^s)$.[10]

Clearly, these conditions are highly unlikely to occur simultaneously. In fact, in the limit as a_i^s approaches a_{i+1}^s, the likelihood of finding any fidelity value Φ_ψ^s in the interval $[a_i^s, a_{i+1}^s)$ approaches zero. Moreover, if no fidelity value Φ_ψ^s is found in $[a_i^s, a_{i+1}^s)$ during frame slot s, then $(a_{i+1}^s - a_i^s)$ increases by the amount $[R(d_{i+1}) - R(d_i)]$, in accordance with (16), so that threshold misordering becomes even less likely in future frame slots. Thus, when the adaptation time constant B is large, so that the relative change in fidelity threshold values from frame slot to frame slot is very small, many increasingly unlikely events must occur in succession in order to gradually bring a pair of fidelity thresholds sufficiently close to allow misordering to occur. In fact, in Section VI we demonstrate experimentally that the likelihood of order violations decreases so rapidly as B increases that they may never be observed in a practical application. Nevertheless, the adaptive strategy described so far does admit the possibility of order violations. To avoid this possibility altogether, the adaptive algorithm may detect an impending violation and then slightly modify the reference distortion values so as to prevent the order violation. To be precise, in some frame slot s, suppose that $(a_{j+1}^s - a_j^s) > 0$, $\forall j$, but the adaptation algorithm of (10) is about to produce an order violation, $(a_{i+1}^{s+1} - a_i^{s+1}) \le 0$, for some i. The violation may be avoided by artificially modifying those fidelity values Φ_ψ^s which lie in the interval $[a_i^s, a_{i+1}^s)$, moving them outside the interval. Naturally, this artificial modification of the fidelity values, which is equivalent to an artificial modification of the original reference distortion values, interferes with the distortion properties of constant distortion scaling. The modification, however, need only be very slight because an impending order violation requires $[a_i^s, a_{i+1}^s)$ to be a relatively small interval. Moreover, according to (15), the map $\mathcal{M}$, from reference distortion values to fidelity values, is effectively scaled by the time constant, B. This means that the amount by which any reference distortion value must be changed in order to move the corresponding fidelity value outside the interval $[a_i^s, a_{i+1}^s)$ decreases in inverse proportion to the size of B.

If the smallest fidelity value in the interval $[a_i^s, a_{i+1}^s)$ is moved immediately below[11] a_i^s then $\psi^s(d_i)$ is incremented by one and (10) assigns a_i^{s+1} a slightly smaller value than it otherwise would have; according to (9), however, the values of $\psi^s(d_j)$, $j \ne i$, are unaffected by this modification. On the other hand, if the largest fidelity value in the interval $[a_i^s, a_{i+1}^s)$ is increased to a_{i+1}^s, then $\psi^s(d_{i+1})$ is decremented by one and (10) assigns a_{i+1}^{s+1} a slightly larger value than it otherwise would have. Again, because $a_{i+2}^s > a_{i+1}^s$, this modification will not affect the values of $\psi^s(d_j)$, $j \ne i+1$. In this way, by slightly modifying the fidelity values $\Phi_\psi^s \in [a_i^s, a_{i+1}^s)$, we are able to independently increment $\psi^s(d_i)$ and/or decrement $\psi^s(d_{i+1})$, causing $[\psi^s(d_{i+1}) - \psi^s(d_i)]$ to decrease and $(a_{i+1}^{s+1} - a_i^{s+1})$ to increase, so that order violations may be avoided. Because this approach gives such precise control over the values of $\psi^s(d_i)$ and $\psi^s(d_{i+1})$, it is possible to show that a suitable choice always exists, which does

[8] To see this, observe that $\Phi_{\psi^s(d_i)} \ge a_i^s$, from (9). But, if we also have $\Phi_{\psi^s(d_i)} \ge a_{i+1}^s$, then (9) yields $\psi^s(d_{i+1}) = \psi^s(d_i)$, contradicting $\psi^s(d_i) < \psi^s(d_{i+1})$.

[9] By small, we mean in relation to the overall range of fidelity values. Assuming that adaptation is slow, which it must be if distortion is to be held constant over a reasonable period of time, the relative change in any fidelity threshold value between frame slots must also be slow, which means that $(a_{i+1}^s - a_i^s)$ must be very small if these thresholds are to cross in the next frame slot, $s+1$.

[10] Of course, these conditions are necessary, but not sufficient. In fact, we may require several substreams, ψ, to have fidelity values in the narrow interval, $[a_i^s, a_{i+1}^s)$, in order for $\sum_{\xi=\psi^s(d_i)+1}^{\psi^s(d_{i+1})} R_\xi$ to exceed $(a_{i+1}^s - a_i^s) + [R(d_{i+1}) - R(d_i)]$.

[11] By this we mean that $\Phi_\psi^s \in [a_i^s, a_{i+1}^s)$ is reduced just sufficiently to guarantee that $a_{i-1}^s < \Phi_\psi^s < a_i^s$ and $\Phi_\psi^s > \Phi_{\psi-1}^s$.

not violate the bounds in Observation 3 and therefore does not disturb any of the properties of our adaptation scheme discussed hitherto [26].

So far, we have had to resort to artificial modification of the reference distortion values V_ψ^s on two separate occasions, in order to guarantee that the leaky bucket criterion of (7) is satisfied: first to enforce the bounds $V_\psi^{\min} \le V_\psi^s \le V_\psi^{\max}$; and second to guarantee the fidelity threshold ordering $a_1^s < a_2^s < \cdots < a_p^s$. In many applications, it may also be desirable to offer hard guarantees on the maximum and/or minimum bit rates associated with some distortion target, $\mathcal{D} = d_i$, i.e., $\psi^{\min}(d_i) \le \psi^s(d_i) \le \psi^{\max}(d_i), \forall\, s$. Such guarantees may be accommodated using the same approach. For example, we might use a collection of representative video sequences to identify values $\psi^{\min}(d_i)$ and $\psi^{\max}(d_i)$, such that we expect to find $\psi^{\min}(d_i) \le \psi^s(d_i) \le \psi^{\max}(d_i)$ with a high degree of confidence. Due to the unpredictable nature of interactive video material, however, it is possible that these bounds might occasionally be violated. To avoid the possibility of such violations, we want to make sure that $\psi^s(d_i) = \psi^{\min}(d_i)$ or $\psi^s(d_i) = \psi^{\max}(d_i)$, as appropriate, in any frame slot s, where such a violation would otherwise occur. This hard-limiting of the $\psi^s(d_i)$ values may be accomplished by appropriately modifying the distortion tag values $\mathcal{D}_\psi^s$, which, in turn, is equivalent to appropriately modifying the reference distortion values, V_ψ^s, prior to applying the algorithm presented above. It is not difficult to see that such a modification need not disturb the fidelity threshold bounds of Observation 3, so that the leaky bucket criterion of (7) is still satisfied.

3) Summary of Proposed Distortion Tagging Algorithm: In summary, we have described an algorithm for adapting the map $\mathcal{T}^s$ from reference distortion values, V_ψ^s, to distortion tag values, $\mathcal{D}_\psi^s$, such that $\mathcal{T}^s$ depends only upon the past, i.e., the values V_ψ^{s-1}, V_ψ^{s-2}, $\cdots$. In order to guarantee the leaky bucket criterion of (7) and possibly other hard constraints on the instantaneous bit rate, we must occasionally make artificial modifications to the actual reference distortion values, V_ψ^s. Because these modifications occur rarely and involve only small changes, we do not expect them to significantly impact the distortion properties of constant distortion scaling. Moreover, the V-distortion measure itself can, at best, be thought of as an approximate measure of actual subjective distortion. Of greater concern is the fact that the need to adapt $\mathcal{T}^s$ means that there is a limited time frame, over which we may consider the distortion to be held approximately constant. This time frame is determined by the parameter, B. Larger values for B allow the distortion to be held approximately constant over longer periods of time, but lead to slower response times in policing the leaky bucket model of average bit rate.

It is helpful at this point to summarize the elements of our proposed adaptation scheme. The first step is for all elements of the compression, storage, and distribution path associated with the compressed video to agree on a set of fixed parameters. These include the set of valid distortion targets, $\{d_1, d_2, \cdots, d_p\}$, the standard average rate function, $R(\cdot)$, the leaky bucket time constant, B, and possibly also a set of rate bounds, $\psi_i^{\min}$ and $\psi_i^{\max}$, $i = 1, 2, \cdots, p$. These parameters establish the context within which scaling and regulating entities should interpret the distortion tag values embedded in the layered substream hierarchy. In Section VI-A we describe the specific parameter choices adopted for our experimental investigations. In addition to these commonly agreed parameters, the adaptation algorithm must also place upper and lower bounds, $V_\psi^{\max}$ and $V_\psi^{\min}$, on the reference distortion values. Because these bounds must be known ahead of time, they should probably be obtained from the statistics of some collection of representative video sequences. During the distortion tagging process itself, these bounds are strictly enforced by hard-limiting the actual reference distortion values if necessary. The fixed map, $\mathcal{M}$, may then be designed to satisfy (15) using the simple piecewise linear approach described above, or any other suitable method. The distortion tagging operation proceeds by first mapping reference distortion values V_ψ^s to fidelity values Φ_ψ^s through $\mathcal{M}$, and then mapping the fidelity values Φ_ψ^s to distortion tags $\mathcal{D}_\psi^s$ through $\mathcal{A}^s$, which is updated after each frame slot, according to (10). As discussed above, the fidelity values, or equivalently, the reference distortion values, may occasionally need to be modified slightly in order to prevent misordering of the thresholds, $a_1^s < a_2^s < \cdots < a_p^s$, of $\mathcal{A}^s$. However, both the frequency and the magnitude of such modifications both decrease rapidly as the time constant B becomes large.

IV. Highly Scalable Compression

The purpose of this section is to discuss video compression algorithms, which are able to generate highly scalable compressed data streams, conformable to the layered substream abstraction introduced in Section II. In Section I, we referred to a number of existing scalable compression schemes. Since predictive coding techniques such as the popular motion compensation approach adopted by the H.261, MPEG-1, and MPEG-2 standards, are not well-suited to highly scalable compression, we must resort to some form of 3-D transform as a means of exploiting both spatial and temporal redundancy. It is important to realize, however, that a multiresolution transform does not in itself guarantee that the compression scheme will be highly scalable. Although simply discarding various resolution components from a multiresolution decomposition provides a mechanism for scaling the resolution and computational demands associated with video decompression, it is inadequate for many applications, particularly those requiring bit rate scalability. There are two reasons for this. First, such an approach offers only very coarse control over the bit rate [19]. More importantly, however, quantization noise in the lower resolution components of a multiresolution decomposition is much more noticeable when these components are to be used in reconstructing the video sequence at high resolution than it is when only a low resolution picture is to be reconstructed. As a consequence of this phenomenon, any compression scheme which employs a multiresolution transform but offers only a single level of quantization for each multiresolution component generally operates at an unsuitable rate-distortion point for all but at most one of the available decompression resolutions [13]. In conclusion, we should expect a successful

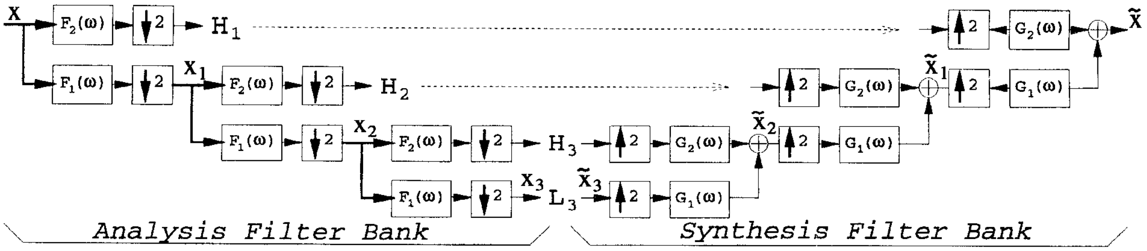


Fig. 3. $L = 3$ level, 1-D multiresolution transform with diadic decomposition at each level.

highly scalable video compression scheme to involve a 3-D multiresolution transform as well as a layered quantization and coding scheme to provide multiple levels of quantization for each resolution component. We refer to this latter task as *progressive* quantization and coding because it permits progressive refinement of the accuracy to which each resolution component is recovered as the number of available quantization layers increases. Suitable multiresolution transforms are discussed in Section IV-A, while progressive quantization and coding approaches are considered in Section IV-B.

A. Multiresolution Transforms

The multiresolution transforms considered in this paper are obtained by separable application of one-dimensional (1-D) filtering and subsampling operations along the temporal and spatial dimensions. Fig. 3 provides an example of an $L = 3$ level 1-D multiresolution transform. In the figure, F_1 and F_2 represent low and high pass analysis filters, respectively, while G_1 and G_2 represent low and high pass synthesis filters. The operator, $\boxed{\downarrow 2}$, denotes subsampling by a factor of two, i.e., discarding every second sample, while the operator, $\boxed{\uparrow 2}$, denotes up-sampling by a factor of two, i.e., inserting a zero valued sample between every pair of input samples. Together, the filtering and subsampling operators of the analysis system depicted in Fig. 3 decompose the input signal x into a collection of so-called subbands, denoted H_1, H_2, H_3, and L_3 with the same number of samples as the original input signal. If the reconstructed signal, $\tilde{x}$, is identical to the input signal x, up to a translational offset, the combined analysis and synthesis filter banks are said to constitute a perfect reconstruction (PR) subband system. Although PR is a desirable property, near-perfect reconstruction (NPR), for which subband synthesis is only approximately the inverse of subband analysis, is often sufficient in practice. The actual design of suitable analysis and synthesis filters is not discussed here; however [27] provides a useful reference in this area.

We refer to the decomposition of Fig. 3 as a *multiresolution* transform because reconstruction of the signal from a partial collection of subbands is analogous to the conventional concept of resolution scaling. For example, if the filters, F_1, F_2, G_1, and G_2 are selected so as to ensure the PR property, then discarding the H_1 subband and applying only the first two levels of the synthesis filter bank yields the signal, $\tilde{x}_1 = x_1$; the same signal is obtained directly from x by lowpass filtering and subsampling, which are precisely the operations required for resolution reduction. In the same way, successively lower resolution signals, $\tilde{x}_1 = x_1$, $\tilde{x}_2 = x_2$, and $\tilde{x}_3 = x_3$, may all be recovered by discarding a sufficient number of high frequency subbands and partially applying the complete synthesis filter bank. In general, each level of decomposition provides one new potential reconstruction resolution. In diadic decompositions, i.e., those in which each level divides the signal into only two subbands, each successive resolution is related to the previous one by a factor of two. Multiple band decompositions are also possible at each level, in which case the available resolutions are more widely separated. The popular length 8 DCT (discrete cosine transform), for example, may be understood as a subband system which divides the signal into eight subbands in only one level. In the sense described above, this DCT permits reconstruction at only one lower resolution with $\frac{1}{8}$ of the full resolution, which is obtained by discarding all but the DC coefficients of the DCT. The most natural extension of the 1-D transform shown in Fig. 3 to two spatial dimensions is illustrated in Fig. 4. Only the analysis system is actually shown, together with the spectral regions occupied by the various two-dimensional (2-D) subbands in this two level decomposition. In this case, the image is divided into four subbands in each level by separable application of 1-D low and high pass filters and subsampling operators. Again, each level in the decomposition contributes one new potential reconstruction resolution. Such multiresolution image decompositions were initially proposed by Mallat [12].

A full 3-D multiresolution video transform may be obtained by employing the 1-D transformation of Fig. 3 to temporally decompose each spatial subband of Fig. 4. Equivalently, a video sequence may first be subjected to a 1-D temporal transform, after which each temporal subband is spatially decomposed. Ohm [15], Singh *et al.* [19], and Taubman and Zakhor [24] have all proposed such separable 3-D multiresolution transforms for scalable video compression. These authors all propose application of one or more levels of the so-called Haar wavelet transform for the temporal dimension. This PR transform is obtained by employing the simplest possible analysis and synthesis filters, F_1, F_2, G_1, and G_2, in Fig. 3, with two tap impulse responses

$$\begin{aligned} f_1[n] &= g_1[-n] \\ &= \frac{1}{\sqrt{2}} \begin{cases} 1, & \text{if } n \in [0,\, 1] \\ 0, & \text{otherwise} \end{cases} \\ f_2[n] &= g_2[-n] \\ &= \frac{1}{\sqrt{2}} \begin{cases} (-1)^n, & \text{if } n \in [0,\, 1] \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{17}$$

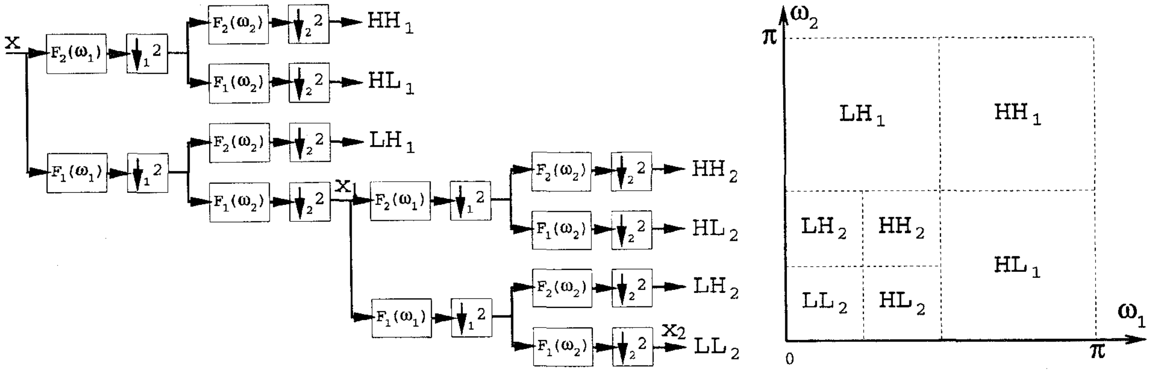


Fig. 4. Two-level, 2-D, separable multiresolution transform.

It is not difficult to see that an L-level temporal Haar wavelet transform may be implemented by independently transforming successive blocks of 2^L video frames each. Consequently, we refer to this transform as block-based. As already mentioned in Section II, this property is useful in minimizing delay and memory demands associated with the transform. In fact, it is not difficult to show [26] that only $L+1$ frame buffers are required to implement a separable 3-D multiresolution transform in which the temporal decomposition is accomplished using an L-level Haar wavelet transform. In addition to these memory and delay considerations, there does not appear to be any compression advantage to using more complex analysis and synthesis filters in the temporal dimension [15], [24].

The compression gain associated with multiresolution transforms such as those discussed above arises principally from the fact that most of the signal information is concentrated in relatively few resolution components,[12] or subbands, a phenomenon known as *energy compaction*. The compression gain associated with temporal subband decomposition depends upon temporal smoothness of the video sequence to concentrate the video information in the lowpass temporal bands. Unfortunately, however, spatial subband transformation is not a shift invariant operation, so that even small amounts of motion in the original video sequence can result in very significant differences between subband coefficients, which have the same spatial coordinates in successive video frames. Consequently, the separable application of a temporal transformation to these spatial subbands offers little if any compression gain except in regions where the video sequence may be considered temporally stationary. To overcome this difficulty, we propose invertibly predistorting the video sequence in such a way as to increase temporal correlation without degrading the quality of the final reconstructed video.

Fig. 5 illustrates this concept in the specific case of a pan compensating pre-distortion. The figure portrays four frames of a hypothetical video sequence, containing no actual scene motion, in which the camera pans to the right at a constant rate of one pixel per frame. For clarity, only one spatial dimension is represented. 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 in each of the frames. In particular, after relabeling, the pixel indices associated with frame 0 run from 0–12, those of frame 1 run from 1–13, and so on. After this relabeling, pixels with the same spatial index in successive video frames are highly correlated and are thus readily compressed within the context of the separable 3-D multiresolution transform. On the other hand, because the 3-D support of the video sequence is no longer rectilinear after relabeling, special care must be taken to preserve the PR property of the multiresolution transform at the boundaries of this region of support. These issues are discussed more thoroughly in [24]. In practical applications, the success of camera pan compensation turns out to be highly dependent on our 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 at most $\pm\frac{1}{2}$ pixel in each of the horizontal and vertical directions. 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, it should be possible to invert this interpolative shift during decompression, as suggested by the *Inverse Pan Compensation* block of Fig. 5. A suitable approach to invertible interpolative shifting has been developed in [23]; this same approach is adopted for all our experimental work.

The pan compensation technique described above is illustrative of a potentially broad class of motion compensated multiresolution transforms, in which the complete transform is considered as the composition of an invertible predistortion and a separable transform. Ohm [15] has proposed a highly successful motion-compensated 3-D subband transform, which may be understood as applying the global pan compensation strategy above to local image blocks rather than the entire frame. There are many other ways, however, in which to generalize the concept. Firstly, small rotations may be accurately approximated by horizontal and vertical skewing of the individual frames. Reference [23] proposes invertible predistortion techniques to compensate for horizontal and vertical skews in image compression applications; however, the same approach may be applied to compensate for small rotations between successive frames for video compression. In

[12] Differing sensitivities of the human visual sensitivities to the various resolution components can also be exploited in compression.

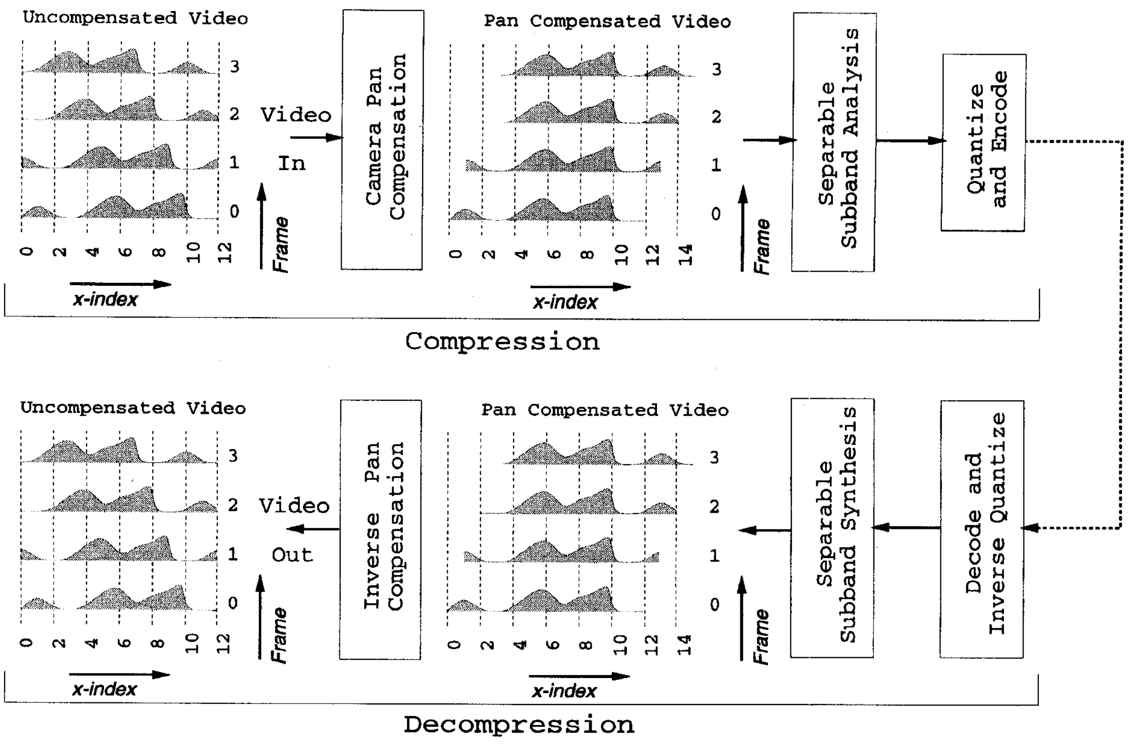


Fig. 5. Illustration of a pan compensated, 3-D multiresolution transform, formed by combining invertible pan compensating predistortion with a separable 3-D transform. For clarity, only one spatial dimension is shown.

this way, both translation and rotation may be compensated, either globally or on a block-by-block basis *a la* Ohm. One could also conceive of employing more general image warping techniques, such as the triangular mesh method described in [14], for example, to compensate for more arbitrary "smooth" motion fields between successive frames.[13] Although we are not concerned with developing such ideas here, the important point to observe is that it is possible to compensate for various types of motion within the context of a 3-D multiresolution transform, without resorting to nonscalable predictive coding techniques such as motion compensated prediction.

B. Progressive Quantization and Coding

While multiresolution transforms are clearly useful for achieving resolution-scalability, our key concern in this paper is with rate-scalability, as explained in Section II. As already mentioned, highly scalable compression requires not only a suitable multiresolution transform, but also an efficient layered quantization and coding scheme for each resolution component or subband. Rather than being forced to select between either discarding or retaining each resolution component during scaling, a layered quantization and coding scheme offers several different operating points on the rate-distortion curve for each resolution component or subband. The purpose of this section is to briefly discuss layered quantization and coding approaches. Section V then deals with the problem of organizing the various quantization layers of each subband into fixed rate substreams with the properties required by our layered substream abstraction, as outlined in Section II.

In general, we have a set of N successively finer quantizers, $\mathcal{Q}_1^b, \cdots, \mathcal{Q}_N^b$, for each subband, b, with associated quantization layers, $\mathcal{L}_1^b, \cdots, \mathcal{L}_N^b$. Quantizer $\mathcal{Q}_n^b$ operates on the samples in subband b to produce a sequence of quantization symbols, $s_n^b[k]$. The symbols $s_1^b[k]$, corresponding to the coarsest quantizer, are coded into layer $\mathcal{L}_1^b$, while the additional information required to recover the symbols $s_n^b[k]$, given that the symbol sequences $s_1^b[k]$, $s_2^b[k]$, $\cdots$, $s_{n-1}^b[k]$ are already available, is coded into layer $\mathcal{L}_n^b$. In this way, the first n quantization layers for subband b are sufficient to reconstruct the subband samples to precision $\mathcal{Q}_n^b$. Thus, the precision to which the subband sample values are recovered is successively refined as the number of available quantization layers increases. Each quantizer $\mathcal{Q}_n^b$ is characterized by its set of Voronoi regions, $\mathcal{V}_n^b$. As a first observation, we may assume, without any loss of generality, that the quantizers are embedded. That is, each Voronoi region, $v \in \mathcal{V}_n^b$, is a subset of one of the Voronoi regions, $v' \in \mathcal{V}_{n-1}^b$. To understand why this assumption causes no loss of generality, suppose that $\mathcal{Q}_1^b, \cdots, \mathcal{Q}_N^b$ are arbitrary quantizers. Then the first n quantization layers in subband b together identify the region $v \in \cap\mathcal{V}_n^b$, to which the subband samples belong, where we define $\cap\mathcal{V}_n^b \triangleq \{v_1 \cap v_2 \cap \cdots \cap v_n \,|\, v_i \in \mathcal{V}_i^b\}$, the set of all regions formed by taking intersections of the Voronoi regions associated with the first n quantizers. Moreover, the quantization layer $\mathcal{L}_n^b$ may be understood as conveying the

[13] To accommodate expansions and contractions we may need to abandon the requirement of exact *invertibility* or else permit some sample rate expansion; however, these might not necessarily be serious drawbacks.

additional information required to specify the particular region in $\cap\mathcal{V}_n^b$ to which the subband samples belong, given that we already know the outcome of the first $n-1$ quantization stages. Thus, we could replace quantizer $\mathcal{Q}_n^b$ with a new quantizer, say $\mathcal{Q}_n'^b$, whose Voronoi regions are those in $\cap\mathcal{V}_n^b$, without changing either the distortion achievable by decoding the first n quantization layers or the additional information which must be specified by quantization layer $\mathcal{L}_n^b$.[14] So, from an information theoretic point of view, it makes no difference whether we consider an arbitrary set of quantizers, $\mathcal{Q}_1^b$, $\mathcal{Q}_2^b$, $\cdots$, $\mathcal{Q}_N^b$, or the set of embedded quantizers, $\mathcal{Q}_1^b$, $\mathcal{Q}_2'^b$, $\cdots$, $\mathcal{Q}_N'^b$, derived by taking the Voronoi regions for quantizer $\mathcal{Q}_n'^b$ to be those in $\cap\mathcal{V}_n^b$, for each n.

The principle question with which we are concerned for progressive quantization and coding schemes is thus how a collection of embedded quantizers, $\mathcal{Q}_1^b$, $\cdots$, $\mathcal{Q}_N^b$, should be selected and their outputs coded so as to yield an efficient compression scheme. For the specific case of an independent identically distributed (IID) Gaussian source with the mean squared distortion measure or an IID Laplacian source with the absolute error distortion measure, Equitz and Cover [7] have shown that it is possible to find embedded vector quantizers to satisfy any arbitrary set of distortion or rate constraints, such that each quantizer approaches optimal rate-distortion performance as its vector length approaches infinity. Such theoretical results are encouraging and provide some motivation for the tree structured vector quantization (TSVQ) schemes proposed by Ohm [15] and Singh *et al.* [19] for scalable video compression. Unfortunately, however, statistical independence is usually a very poor model for subband sample values; moreover, practical vector lengths for vector quantizers are usually very limited. For these reasons, both Singh *et al.* and Ohm observe gradual degradation in compression performance as the number of layers in their embedded quantization and coding schemes is increased.

An alternative approach to TSVQ is to combine embedded scalar quantization with conditional entropy coding to exploit the mutual statistical information associated with the resulting scalar quantization symbols. In fact, an analysis by Gao *et al.* [9] suggests that scalar quantizers are particularly well suited to coding subband sample values. Fig. 6 illustrates the Voronoi regions, or quantization intervals, associated with a particularly useful set of embedded scalar quantizers for high frequency subband samples, which tend to be clustered about their mean value of zero. Each of these so-called dead zone quantizers, $\mathcal{Q}_n^b$, is characterized by a step size, δ_n^b, and a dead zone of size υ_n^b, which is centered about zero. It is not difficult to see that a set of embedded dead zone quantizers must satisfy $\delta_n^b = \delta_{n-1}^b/K_n^b$, and $\upsilon_n^b = \upsilon_{n-1}^b - 2K_n^{b\prime}\delta_n$, where $K_n^b > 0$, and $K_n^{b\prime} \geq 0$ are integers. In the example of Fig. 6, $K_n^b = 2$, and $K_n^{b\prime} = 1$. This selection is particularly attractive from an implementation perspective, because the entire set of quantizers may be realized simply by discarding least significant bits in an appropriately scaled, sign-magnitude representation of the subband samples. Moreover, the analysis in [22] suggests that these scalar quantizers should be approximately optimal in the rate-distortion sense, provided the subband samples conform to a Laplacian distribution; in fact, high frequency subband samples often do exhibit an approximately Laplacian distribution [29].

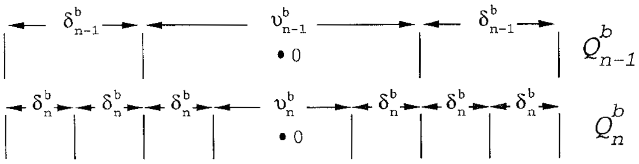


Fig. 6. Two layers of embedded scalar dead zone quantization.

In order for layered coding with embedded scalar quantizers to be efficient, it is necessary to carefully exploit statistical dependencies both between the quantization symbols produced by any given subband sample in successive quantization layers and also between the quantization symbols produced by different subband samples. Conditional arithmetic coding provides a powerful tool for exploiting these dependencies. In this case, each symbol, $s_n^b[k]$ produced by quantizer $\mathcal{Q}_n^b$ is coded with respect to a conditioning context, $\kappa_n^b[k]$, which depends only on those symbols which we can be certain the decoder has already received. In a practical implementation, $\kappa_n^b[k]$ forms an index into a table of conditional statistical distributions for $s_n^b[k]$, which are obtained either by training the coder on a suitable ensemble of source video material in advance or by adapting the conditional distributions on the fly, based on previous occurrences of the various context values. The value of $s_n^b[k]$ is then arithmetically coded [21] using the conditional distribution indicated by $\kappa_n^b[k]$. The success of such a scheme depends upon careful design of the conditioning contexts, $\kappa_n^b[k]$, so as to capture as much information as possible concerning the statistical dependencies between $s_n^b[k]$, and previously coded quantization symbols, while keeping the set of potential context values and hence the size of the statistical tables within manageable bounds.

In [24], we propose an efficient layered coding system for 3-D multiresolution transforms. In this scheme, the conditioning context, $\kappa_n^b[k]$, is based on: 1) the quantization symbol for the same subband sample in the previous quantization layer, i.e., $s_{n-1}^b[k]$; 2) the quantization symbols for spatially adjacent samples from the same subband in the same and previous quantization layers, which have already been coded; and 3) quantization symbols for spatially and temporally coincident samples in different subbands. The contexts are formed using only bitwise logical operations, with each context variable $\kappa_n^b[k]$ taking on at most 268 different values. The layered coding scheme is shown to offer excellent rate-distortion performance with a relatively large number of layers, $N \approx 8$, for each subband. Moreover, if quantization symbols from other subbands—item 3) above—are ignored during context formation, the number of potential states for each context variable is reduced to 67, with less than 6% increase in the overall bit rate for a given level of distortion [26]. An extension to this layered coding scheme is proposed in [25], in which interlayer and spatial neighbor conditioning—items 1) and 2) above—are supplemented by temporal conditioning. Tempo-

[14] This is because this new quantizer does not alter the set of all regions formed by taking intersections of the Voronoi regions associated with the first n quantizers.

ral conditioning is accomplished by including quantization symbols from spatially coincident and neighboring samples in the previous frame of subband b in the expression for $\kappa^b_n[k]$. We refer to this extension as interframe coding, because it enables temporal statistical dependencies to be exploited during coding. In fact, if the motion compensating predistortion operator discussed in Section IV-A is effective, then we can expect significant levels of temporal redundancy. Interframe coding is particularly useful for delay sensitive applications, where delay constraints often prohibit the use of more than one or two levels of temporal subband decomposition in the multiresolution transform [25]; efficient compression then depends partially on our ability to exploit temporal redundancy during conditional arithmetic coding.[15] The obvious drawback of interframe coding is that it imposes requirements on the number of quantization layers available at the decoder in the previous frame, before the conditioning context may be correctly formed to decode the quantization symbols in the current frame. Specifically, the formulation proposed in [25] expects quantization layers one through n to be available in the previous frame before the nth layer can be decoded in the current frame of the same subband. This means that the number of useful quantization layers available at the decoder cannot increase from frame to frame, which clearly works against our goal of scalability. To avoid this difficulty, each subband must occasionally be coded using an intraframe coding technique such as that described in [24]. The implications of interframe coding for scalability are considered further in Section V.

V. Generation of Layered Substreams

In this section, we turn our attention to the organization of progressively coded subband samples into the layered substreams of Fig. 1. As presented in Section IV-B, each quantization layer $\mathcal{L}^b_n$ contains the additional information required to reconstruct all samples of subband b at quantization precision Q^b_n, given that the previous $n-1$ quantization layers for subband b have already been decoded. We begin by pointing out that arithmetic coding generates a single indivisible code word to represent the entire collection of source symbols coded. If conditional arithmetic coding is used to generate $\mathcal{L}^b_n$, as discussed in Section IV-B, then we must first partition the samples of subband b into smaller units in time and/or space, generating a separate arithmetic code word for each such unit, or *code block*. This is clearly necessary if we are to scale the number of quantization layers available to the decoder in a time-varying manner. Moreover, the fact that arithmetic encoding and decoding are inherently serial computational tasks means that parallel computational techniques, which are often required to achieve real time video compression/decompression, can only be exploited when a sufficient number of independent code blocks are present at any point in time. For our experimental work, each subband's samples are partitioned into an integral number of rectangular code blocks within each frame, with no more than 6000 samples each, regardless of the frame size. Each block's samples are represented by an individual arithmetic code word, by applying the conditional arithmetic coding techniques discussed in Section IV-B to each block of samples independently, as though the code block boundaries were frame boundaries. By limiting the number of samples assigned to each code block, we limit the maximum decoding time for each block, which is an ideal situation for parallel hardware or software realizations of the compression scheme. Note that block-based quantization and coding schemes such as TSVQ induce a natural code block structure, if used instead of scalar quantization with arithmetic coding.

In our proposed organization, each of the substreams of Fig. 1 contains the code bits corresponding to an integral number of quantization layers from each subband code block, so that the first ψ substreams in frame slot s collectively represent the first $n^\beta_\psi(s)$ quantization layers of code block β in every frame of frame slot s.[16] Our task, then, is to describe a *rate limiting* algorithm, whose function is to select suitable values, $n^\beta_\psi(s)$, in each frame slot, s, such that the total number of bits required for these quantization layers, together with auxiliary syntactic constructs, does not exceed $\sum_{\xi=1}^{\psi} R_\xi\,(\mathcal{F}/F_R)$. To make this statement more precise, let $B^\beta_n(s)$ denote the total number of code bits and auxiliary header bits required at the decoder in order to unambiguously decode the first n quantization layers of code block β in every frame of frame slot s. Our objective is to select $n^\beta_\psi(s)$ values for every frame slot, s, code block, β, and substream, ψ, such that

$$\sum_{\beta} B^\beta_{n^\beta_\psi}(s) \lessapprox \sum_{\xi=1}^{\psi} R_\xi \frac{\mathcal{F}}{F_R}, \quad \forall \psi,\, s. \tag{18}$$

The selection of the $n^\beta_\psi(s)$ values is additionally constrained in the following two respects: 1) $n^\beta_\psi(s)$ must be at least as large as $n^\beta_{\psi-1}(s)$; and 2) $n^\beta_\psi(s)$ may not exceed $n^\beta_\psi(s-1)$ for any code block whose subband samples are interframe coded in the first frame of frame slot s. This latter requirement arises from the fact that the first n quantization layers of an interframe coded block, β, are decodable only if at least n quantization layers of code block β are available in the previous frame, as described in Section IV-B. The requirement that $n^\beta_\psi(s) \leq n^\beta_\psi(s-1)$ for blocks, β, whose first frame in frame slot s is interframe coded, is necessary to ensure that all code bits contained in the first ψ substreams of frame slot s may be decoded, provided at least ψ substreams were received in frame slot $s-1$. This means that all code bits remaining after constant bit rate substream scaling must be decodable. On the other hand, when constant distortion scaling is employed the number of substreams, $\psi^s(\mathcal{D})$, available in frame slot s may be larger than the number $\psi^{s-1}(\mathcal{D})$ available in frame slot $s-1$, in which case some of the available code bits in frame

[15] Note that context formation and conditional coding do not introduce any inherent delay into the compression scheme, whereas the temporal multiresolution transform does, as discussed at the end of Section II.

[16] Note that we do not allow the number of quantization layers associated with any code block β to vary from frame to frame within a frame slot. This is primarily to minimize the syntactic overhead associated with our scalable data streams.

slot s may not be decodable.[17] The two constraints above may be summarized as follows:

$$n_{\psi}^{\beta}(s) \geq n_{\psi-1}^{\beta}(s), \quad \forall \psi, \beta, s \tag{19}$$

$$n_{\psi}^{\beta}(s) \leq \nu_{\psi}^{\beta}(s), \quad \forall \psi, \beta, s \tag{20}$$

where

$$\nu_{\psi}^{\beta}(s) \triangleq \begin{cases} n_{\psi}^{\beta}(s-1), & \text{if } \beta \text{ interframe coded} \\ & \text{in first frame of slot } s \\ \infty, & \text{otherwise.} \end{cases}$$

The particularly restrictive nature of (20) forces us to resort to intraframe coding of subbands from time to time, as mentioned in Section IV-B. The distribution of such intraframe coding events is indicated in Section VI for particular highly scalable compression algorithms.

Naturally, there are many potential combinations of $n_{\psi}^{\beta}(s)$ values satisfying (18)–(20), among which we would prefer to make the selection which minimizes distortion in the reconstructed video sequence. Useful multiresolution transforms typically correspond to projections onto orthonormal or approximately orthonormal sets of basis vectors, which span the space of all video signals.[18] Thus, it is usually a simple matter to obtain an accurate estimate, $D_n^{\beta}(s)$, for the independent contribution of each code block, β, to the mean squared reconstruction error over frame slot s, when only n quantization layers of block β are decoded in each frame of the frame slot. In theory, the $B_n^{\beta}(s)$, and $D_n^{\beta}(s)$ values may be used to minimize MSE in the reconstructed video sequence subject to (18)–(20). Unfortunately, however, the discrete parameter space renders exact optimization a computationally infeasible task, even for a relatively small number of code blocks. In view of this obstacle, we propose the following rate limiting algorithm.

1) Find the largest value of N_{ψ} such that

$$\sum_{\beta \in K_{N_{\psi}}} B_{\nu_{\psi}^{\beta}(s)}^{\beta}(s) + \sum_{\beta \notin K_{N_{\psi}}} B_{N_{\psi}}^{\beta}(s) \leq \sum_{\xi=1}^{\psi} R_{\xi} \frac{\mathcal{F}}{F_R}$$

and set T equal to the left hand side of the above inequality, where $K_{N_{\psi}}$ is given by

$$K_{N_{\psi}} = \{\beta | \nu_{\psi}^{\beta}(s) \leq N_{\psi}\}.$$

2) Sort the collection of code blocks so that block β_1 precedes block β_2 whenever $n_{\psi-1}^{\beta_1}(s) = N_{\psi} + 1$, and $n_{\psi-1}^{\beta_2}(s) \leq N_{\psi}$ or, failing this, if

$$\frac{D_{N_{\psi}+1}^{\beta_1}(s) - D_{N_{\psi}}^{\beta_1}(s)}{B_{N_{\psi}+1}^{\beta_1}(s) - B_{N_{\psi}}^{\beta_1}(s)} < \frac{D_{N_{\psi}+1}^{\beta_2}(s) - D_{N_{\psi}}^{\beta_2}(s)}{B_{N_{\psi}+1}^{\beta_2}(s) - B_{N_{\psi}}^{\beta_2}(s)}.$$

3) For each code block, $\beta = 1, 2, \cdots$, sorted as above

- If $\beta \notin K_{N_{\psi}}$, and $T + B_{N_{\psi}+1}^{\beta}(s) - B_{N_{\psi}}^{\beta}(s) \leq \sum_{\xi=1}^{\psi} R_{\xi} \mathcal{F}/F_R$, set $n_{\psi}^{\beta}(s) = N_{\psi} + 1$, and $T = T + B_{N_{\psi}+1}^{\beta}(s) - B_{N_{\psi}}^{\beta}(s)$.
- Otherwise, set $n_{\psi}^{\beta}(s) = \min\{N_{\psi}, \nu_{\psi}^{\beta}(s)\}$.

In each frame slot s, the proposed algorithm is applied to substreams $\psi = 1, 2, \cdots, \Psi$ in succession. As we shall see, this algorithm is best understood as an attempt to allocate every code block exactly the same number of quantization layers, N_{ψ}. The appropriateness of this objective depends upon suitable selection of the sets of quantizers associated with each subband. Higher priority may be assigned to lower frequency subbands, b, for example, simply by assigning them finer quantizers, $\mathcal{Q}_n^b$. Step 1 of our proposed rate limiting algorithm finds the maximum value for N_{ψ} such that each code block, β, which is not otherwise constrained by (20), may be allocated $n_{\psi}^{\beta} = N_{\psi}$ quantization layers without violating the rate limit, (18). This first step of the algorithm is expressed in terms of the set $K_{N_{\psi}}$, of all code blocks which may not be allocated more than N_{ψ} quantization layers without violating (20). That is, $K_{N_{\psi}}$ contains those code blocks, β, which may not be allocated more than $\nu_{\psi}^{\beta}(s) \leq N_{\psi}$ quantization layers without violating interframe coding dependencies. In step 1, T is assigned to be the number of bits in frame slot s, required to represent $\nu_{\psi}^{\beta}(s)$ quantization layers of all code blocks, $\beta \in K_N$, and N_{ψ} quantization layers of all remaining code blocks.

Because the fixed substream bit rates R_{ψ} are not generally related to the number of bits generated during layered coding of the code block samples, we cannot expect T to be equal to, or even very close to the limit, $\sum_{\xi=1}^{\psi} R_{\xi} (\mathcal{F}/F_R)$. In order to make better use of available resources, therefore, the second and third steps of our proposed rate limiting algorithm are responsible for selecting some blocks, $\beta \notin K_{N_{\psi}}$, for allocation of an extra quantization layer, i.e., $n_{\psi}^{\beta}(s) = N_{\psi} + 1$. Step 2 establishes an order in which blocks are to be considered for allocation of this extra quantization layer. It can happen that some code block, β, has already been allocated $N_{\psi}+1$ quantization layers in the previous substream, i.e., $n_{\psi-1}^{\beta}(s) = N_{\psi} + 1$, in which case we must select $n_{\psi}^{\beta}(s) = N_{\psi} + 1$ in order to satisfy (19). This is managed by ensuring that such code blocks appear first in the order established during step 2. The remaining code blocks, β, are organized in order of increasing rate-distortion gradient, $[D_{N_{\psi}+1}^{\beta}(s) - D_{N_{\psi}}^{\beta}(s)]/[B_{N_{\psi}+1}^{\beta}(s) - B_{N_{\psi}(s)}^{\beta}]$.[19] This means that the blocks, β, which are allocated an extra quantization layer, are to be those which offer the greatest decrease in reconstruction MSE, $D_{N_{\psi}}^{\beta}(s) - D_{N_{\psi}+1}^{\beta}(s)$, relative to the number of additional bits, $B_{N_{\psi}+1}^{\beta}(s) - B_{N_{\psi}}^{\beta}(s)$, required for this extra quantization layer. Finally, in step 3 of our proposed rate limiting algorithm, code blocks $\beta \notin K_{N_{\psi}}$ are examined one at a time, in the order established during step 2, to be allocated the additional quantization layer, $n_{\psi}^{\beta}(s) = N_{\psi} + 1$,

[17] As we shall see in Section VI, the nature of the variable bit rate traffic generated by constant distortion scaling ensures that the number of undecodable code bits is smaller than one might at first suppose.

[18] This is certainly true for the 3-D transforms employed in [19], [24], and [25] and approximately true for the transforms employed by Ohm in [15].

[19] To avoid confusion here, note that the rate-distortion gradient is always negative. That is, distortion always decreases as the bit rate increases.

TABLE I

SUBSTREAM BIT RATES. R_ψ IS THE RATE OF SUBSTREAM ψ, WHILE $R(-\psi) \triangleq \sum_{\xi=1}^{\psi} R_\xi$ IS THE BIT RATE OF THE SCALABLE DATA STREAM FORMED FROM THE FIRST ψ SUBSTREAMS

ψ	1	2	3	4	5	6	7	8	9	10	11	12	13	14
R_ψ (kbps)	63.36	63.36	63.36	63.36	63.36	63.36	126.72	126.72	126.72	126.72	126.72	126.72	190.08	190.08
$R(-\psi)$ (kbps)	63.36	126.72	190.08	253.44	316.80	380.16	506.88	633.60	760.32	887.04	1013.76	1140.48	1330.56	1520.64

ψ	15	16	17	18	19	20	21	22	23	24	25	26	27	28
R_ψ (kbps)	253.44	253.44	253.44	253.44	380.16	380.16	506.88	506.88	506.88	506.88	760.32	760.32	1013.76	1013.76
$R(-\psi)$ (kbps)	1774.08	2027.52	2280.96	2534.40	2914.56	3294.72	3801.60	4308.48	4815.36	5322.24	6082.56	6842.88	7856.64	8870.40

so long as the limit $\sum_{\xi=1}^{\psi} R_\xi\,(\mathcal{F}/F_R)$ is not exceeded. In this operation, T keeps track of the total number of bits allocated to all code blocks in the frame slot. It is interesting to note that in our experimental investigations, the final value of T is virtually always within 1% of the limit, $\sum_{\xi=1}^{\psi} R_\xi\,(\mathcal{F}/F_R)$.

In the above discussion, we have made no mention of the different roles played by luminance and chrominance component code blocks during rate limiting of color video signals. In fact, the above algorithm is only suitable for monochromatic compressed video. In [26] we describe a modification to this rate limiting approach, for full color compressed video. Importantly, this modified algorithm does not depend on an additive model for the distortion associated with the luminance and chrominance components.

VI. EXPERIMENTAL WORK

In this section, we present experimental findings to indicate the performance of our proposed layered substream abstraction, with both constant bit rate and constant distortion scaling criteria, when used in conjunction with a suitable highly scalable compression scheme. In view of the generality of the material presented in Sections II, III, and IV, it is appropriate that we first offer some specific details of the context in which these experimental results are to be understood. To this end, Section VI-A discusses the specific parameter choices adopted for the layered substream hierarchy itself, while Section VI-B outlines the key features of the highly scalable video compression algorithms used to generate experimental substreams. The actual experimental findings are then presented in Sections VI-C and VI-D.

A. Specific Choices for the Layered Substream Hierarchy

The parameter choices outlined in this section are useful both as a framework within which to understand the experimental results of Sections VI-C and VI-D and as a specific context within which to appreciate the more abstract discussion of distortion tag generation in Section III. For convenience, we select distortion target values from the set $\{-1, -2, \cdots, -\Psi\}$ of negated substream numbers, i.e., $d_i = -i$, $i = 1, 2, \cdots, p = \Psi$, and we define the standard average rate function, $R(\mathcal{D})$, to be

$$R(\mathcal{D}) \triangleq \sum_{\psi=1}^{-\mathcal{D}} R_\psi. \tag{21}$$

This definition has the interesting consequence that an average bit rate of $R(\mathcal{D})$ may be obtained either by constant distortion scaling, in which all but the first $\psi^s(\mathcal{D})$ substreams are discarded in each frame slot, s, or by constant rate scaling, in which all but the first $-\mathcal{D}$ substreams are discarded in every frame slot. In the latter case, of course, the average and instantaneous bit rates coincide. Thus, exactly the same set of average bit rates is available for both constant distortion scaling and constant rate scaling. Note that the standard average rate function, $R(\cdot)$, defined in (21), is also useful for expressing the constant bit rate associated with the first ψ substreams of our layered hierarchy, i.e., $R(-\psi)$. This dual role of the rate function $R(\cdot)$ is exploited in the notation of Table I, which indicates the substream bit rates, R_ψ, and the associated cumulative substream bit rates, $R(-\psi)$, adopted for our experimental investigations. The standard rate function, $R(\cdot)$, defined in (21), has two other important consequences. First, because the minimum average bit rate, $R(d_1)$, is identical to the minimum instantaneous bit rate, R_1, the set of potential distortion tag values and the set of distortion targets are one and the same. That is, the extra distortion tag value, d_0, is superfluous, as explained in Section II. The second consequence of (21) is that the parameters, ψ_i^a, and ψ_i^b, of (11) and (12), satisfy $\psi_i^a = \psi_i^b = -i$, $\forall\, i$. This simplifies (15) and hence construction of the fixed part, $\mathcal{M}$, of the distortion tagging map, as described in Section III-B.

For simplicity, we consider only a simplistic V-distortion measure, which is based around MSE. Recall that the V-distortion measure forms the starting point in distortion tag generation, as discussed in Section III. The multiresolution transform described in Section VI-B effectively projects the source video sequence onto a nearly orthonormal set of basis vectors, which span the space of all video sequences. Consequently, a good approximation, $V_\psi^{s,c}$, to the MSE of any color component, c, over frame slot s, for the video sequence reconstructed from substreams $1, 2, \cdots, \psi$, may readily be obtained by summing MSE contributions from individual subband samples. These MSE contributions may be determined during compression with little computational overhead. The only remaining task is to form a single distortion value, V_ψ^s, from the three color component distortions, $V_\psi^{s,y}$, $V_\psi^{s,u}$, and $V_\psi^{s,v}$. To that end, we, somewhat arbitrarily, adopt the formulation

$$V_\psi^s = V_\psi^{s,y} + \tfrac{1}{4}\,(V_\psi^{s,u} + V_\psi^{s,v})$$

which reflects a view that the chrominance components should have less impact on subjective distortion than the luminance component. This MSE based V-distortion measure is particularly useful for numerically demonstrating the performance of constant distortion substream scaling, even if it does not closely reflect actual subjective distortion.

B. Specific Choices for Highly Scalable Compression

For the investigations here, we adopt five examples from the class of highly scalable compression algorithms described in [25]. These algorithms all employ a separable 3-D multiresolution transform, with four levels of the spatial transform[20] illustrated in Fig. 4 and L levels of the 1-D transform in Fig. 3 applied temporally to each spatial subband; the parameter L takes on values of 1, 2, and 3 in our various example compression algorithms, as discussed shortly. The simple two tap filters of (17) are adopted for the temporal direction so that our temporal transform is an L-level Haar wavelet transform. Recall from Section IV-A that our multiresolution transform is then block-based in time, with a block size of 2^L frames. For spatial subband filters, we adopt the nine tap NPR subband filters of Adelson *et al.* [1], with symmetric extension [20] applied at the frame boundaries to avoid sample rate expansion. These filters are selected because they lead to a nearly orthonormal set of transform basis vectors, which is a useful property when working with the MSE distortion metric, as proposed in Sections V and VI-A. The 3-D multiresolution transform is supplemented by the invertible pan compensating predistortion operator discussed in Section IV-A to improve exploitation of temporal redundancy in scenes exhibiting global translational motion.

From the layered quantization and coding approaches touched upon in Section IV-B, we adopt the embedded scalar quantizers illustrated in Fig. 6, together with the conditional arithmetic coding contexts described in [25]. Interframe coding is applied to the subbands of low temporal frequency, while the subbands of high temporal frequency are only intraframe coded. This is appropriate, in view of the low interframe temporal redundancy typically exhibited by high temporal frequency subbands. Noting that our experimental comparisons in Sections VI-C and VI-D are to be based on MSE, or its derivative, PSNR,[21] *the optimal approach to quantizer* parameter selection, within each color component, is to use exactly the same set of quantizers for every subband[22] [28, sec. 11.2]. To be precise, all luminance subbands b have a base quantization step size of $\delta_1^b = 512$, while all chrominance subbands b have a base quantization step size of $\delta_1^b = 400$. This ratio of luminance to chrominance quantization precision is found empirically to offer approximately the same relative luminance and chrominance distortions as those experienced with the MPEG-1 compression standard. Quantization step sizes in the remaining quantization layers are given by $\delta_n^b = 2^{1-n}\delta_1^b$, $\forall\, n$, b, while the quantizer dead zones are set to twice the corresponding step size, i.e., $\upsilon_n^b = 2\delta_n^b$, $\forall\, n$, b. The arithmetic coding probability tables discussed in Section IV-B are obtained by training the compression scheme using the three ISO standard test sequences, "pingpong," "football," and "flower garden," at SIF525 resolution.[23]

As mentioned already, we consider five examples from the class of compression schemes outlined above; these have the parameters shown in Table II. The individual algorithms in this table are distinguished on the basis of the number of levels of temporal subband decomposition, L, the number of frames, $\mathcal{F}$, in each frame slot and the intraframe coding interval, $\mathcal{I}$, for subbands of low temporal frequency, as shown in the first three columns of Table II. Although the low temporal frequency subbands are generally interframe coded, the need to occasionally intersperse frames which are purely intraframe coded has already been established in Section V. We adopt the following policy. Within each frame slot, all but the first frame of each low temporal frequency subband are coded using interframe techniques to exploit temporal redundancy; the first frame is also interframe coded, except in every $\mathcal{I}$th frame slot, where intraframe coding alone is employed. Larger values of $\mathcal{I}$ allow higher compression efficiency during periods of reasonably constant scene activity, but can excessively constrain the bit rate limiting algorithm of Section V when scene activity is highly variable. To understand this, consider the disruptive effect of a scene change during frame slot s, and assume that a fixed number, ψ, substreams is available for decompression in each frame slot, i.e., constant bit rate scaling. Because the scene change in frame slot s reduces compression efficiency, we expect that the number of quantization layers, $n_\psi^\beta(s)$, associated with each code block, β, in the first ψ substreams of frame slot s should be less than the corresponding number of layers, $n_\psi^\beta(s-1)$, in the previous frame slot. For those code blocks, β, associated with low temporal frequency subbands, the number of layers, $n_\psi^\beta(s+1)$, $n_\psi^\beta(s+2)$, $\cdots$, in subsequent frame slots may not increase above the relatively low value, $n_\psi^\beta(s)$, forced by the scene change in frame slot s, until block β is next intraframe coded. This is a consequence of the layer allocation constraint (20). Thus, smaller values for $\mathcal{I}$ allow reconstructed video quality to recover more quickly from the disruptive effects of the scene change, whereas larger values for $\mathcal{I}$ allow for more efficient compression during reasonably continuous levels of scene activity. The intraframe coding intervals appearing in the last four rows of Table II are found to offer a useful compromise for scenes with moderately varying levels of activity, such as the standard ISO test sequence, "pingpong." The compression algorithm corresponding to the first row of Table II is exceptional in that it involves neither temporal subband decomposition nor interframe progressive coding. This purely intraframe compression algorithm provides us with a useful gauge of the degree to which the remaining four video compression algorithms are able to exploit temporal redundancy.

The fourth column in Table II indicates the inherent end-to-end delay, derived at the end of Section II, based on a video

[20] Actually, we apply all four levels of spatial decomposition to the luminance component of the video signal, but only three levels to the two chrominance components. This is motivated by the fact that the video sequences with which we work already have their chrominance components subsampled by a factor of two in the horizontal and vertical directions.

[21] $\text{PSNR} = 10\log_{10} 255^2/\text{MSE}$.

[22] This comment is based on the fact that our subband filters are normalized so that the subband transformation basis vectors are very nearly orthonormal.

[23] That is, 30 progressively scanned 352 × 240 pixel frames per second, with chrominance components subsampled by two in the vertical and horizontal directions

TABLE II
EXAMPLES FROM OUR CLASS OF SCALABLE COMPRESSION ALGORITHMS. THE THIRD COLUMN INDICATES THE INTRAFRAME CODING INTERVAL FOR SUBBANDS OF LOW TEMPORAL FREQUENCY, EXPRESSED IN TERMS OF FRAME SLOTS

L	$\mathcal{F}$	Intra-Frame Interval, $\mathcal{I}$	Inherent Delay (ms)	Inherent Delay Classification	Frame Buffers
0	1	1	33	minimum	1
1	2	4	100	low	3
2	4	3	233	medium	4
3	8	2	500	high	5
0	16	1	1033	very high	2

frame rate of $F_R = 30$ f/s, while the fifth column suggests a classification based on this delay. The last column in Table II indicates memory requirements, expressed in terms of the number of frame buffers required during compression or decompression. These memory requirements may be understood from the fact that each algorithm requires $L + 1$ frame buffers to implement the multiresolution transform, as mentioned in Section IV; the last four algorithms in Table II require an additional frame buffer to store subband samples from a previous frame, which are used in forming the conditioning contexts, $\kappa_n^b[k]$, required for interframe coding. Note that we do not consider the relatively small amount of memory required for temporary storage of compressed data; nor do we consider any storage required to determine camera pan parameters, as described in [24].

C. Investigation of Constant Rate Scaling

In this section, we investigate the performance of the five compression algorithms listed in Table II in the context of constant bit rate (CBR) scaling, via the simple substream discarding approach discussed in Section II. This part of our investigation is important because it indicates the performance of particular compression algorithms when subjected to the constraints imposed by the proposed layered substream abstraction. These constraints are manifested in (19) and (20). We begin by investigating the opportunity to exchange compression performance for system memory requirements when end-to-end delay is not of great concern. To this end, we consider the algorithms listed in the first, fifth, and fourth rows of Table II, which require 1, 2, and 5 frame buffers, respectively. The rate-distortion curves of Fig. 7 indicate overall luminance PSNR values, associated with the SIF525 resolution "pingpong" sequence, when reconstructed from the compressed data streams generated by these three algorithms, after CBR substream scaling. Fig. 7 is particularly interesting because it indicates the degree to which our interframe progressive coding scheme is able to exploit temporal redundancy. The compression algorithm corresponding to the last row of Table II exploits temporal redundancy by interframe coding alone, having $L = 0$. It is interesting that the compression performance of this algorithm appears to be approximately intermediate between that of pure intraframe compression, corresponding to the first row of Table II and that of the algorithm listed in the fourth row of Table II, which employs both interframe coding and $L = 3$ levels of temporal subband decomposition to exploit temporal redundancy.

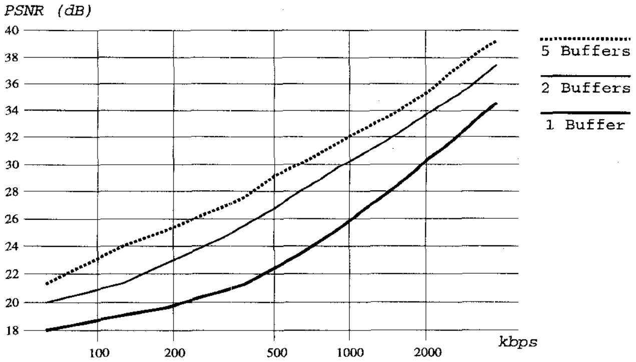


Fig. 7. Luminance PSNR of "pingpong" sequence, reconstructed after CBR substream scaling, using the first, fourth, and fifth algorithms of Table II. Curves identified by memory requirements, expressed in terms of frame buffers.

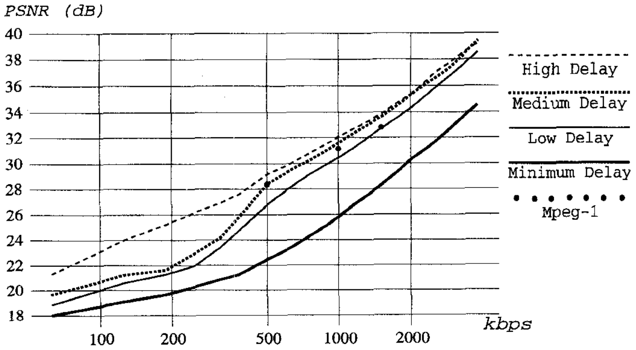


Fig. 8. Luminance PSNR of "pingpong" sequence, reconstructed after CBR substream scaling, using the first four algorithms of Table II. Curves identified by end-to-end delay classification. Specific MPEG-1 PSNR values also shown, for reference.

The algorithms listed in the first four rows of Table II are useful for investigating the opportunity to exchange compression performance for end-to-end delay. The rate-distortion curves of Fig. 8 indicate overall luminance PSNR values, associated with the "pingpong" sequence, when reconstructed from the compressed data streams generated by these four algorithms, after CBR substream scaling. The curves of this figure clearly indicate a law of diminishing returns as delay is exchanged for compression within the framework established by our class of scalable compression algorithms and the proposed layered substream abstraction. Fig. 8 also indicates the luminance PSNR values obtained at three fixed bit rates with an implementation of the nonscalable MPEG-1 compression standard.[24] For reference, the inherent delay associated with this compression algorithm is equal to five frame periods,[25] which falls between the inherent delays of the low and medium delay compression algorithms of Table II. As seen in Fig. 8, the MPEG-1 PSNR figures also fall between those of the

[24] The software MPEG-1 implementation, provided by Bellcore, has the following parameters: 15 frame GOP; 2 B-frames per I- or P-frame; half pixel motion compensation; and a rate control buffer capacity of three frame periods.

[25] Two frame periods, because two B-frames intersperse every pair of I- or P-frames, plus three frame periods from the rate control buffer.

TABLE III
COMPARISON OF MPEG-1 IMPLEMENTATION FROM BELLCORE, WITH LOW AND MEDIUM DELAY ALGORITHMS OF TABLE II. PSNR VALUES FOR MPEG-1 APPEAR IN "MPEG" COLUMNS, WHILE, FOR THE REMAINING ALGORITHMS, IMPROVEMENTS IN PSNR OVER MPEG-1 APPEAR IN THE "LOW" AND "MEDIUM" COLUMNS

Video Sequence	Component	0.5 Mbps			1.0 Mbps			1.5 Mbps		
		MPEG (dB)	Low (dB)	Medium (dB)	MPEG (dB)	Low (dB)	Medium (dB)	MPEG (dB)	Low (dB)	Medium (dB)
'pingpong' 256 frames zoom, pan, still	Y	28.39	-1.57	+0.04	31.15	-0.59	+0.56	32.80	-0.01	+0.88
	U	36.11	-0.79	-0.23	38.19	-0.69	-0.14	39.34	-0.18	+0.39
	V	37.20	-0.17	+0.07	39.16	-0.13	-0.25	40.17	+0.21	+0.50
'football' 160 frames panning	Y	28.62	+0.24	+0.44	31.46	+0.37	+0.53	33.26	+0.79	+1.05
	U	36.22	+1.86	+1.73	38.28	+1.22	+1.26	39.34	+1.15	+1.08
	V	33.34	+2.22	+2.08	35.92	+1.58	+1.58	37.32	+1.65	+1.62
'flower garden' 128 frames translating	Y	22.02	-2.94	-2.09	25.43	-3.68	-2.94	27.16	-3.22	-2.25
	U	30.95	-0.80	-0.27	32.92	-1.92	-1.46	34.12	-2.17	-1.70
	V	27.87	-0.76	-0.47	30.58	-2.48	-2.09	32.25	-2.92	-2.36

low and medium delay scalable compression algorithms, in the case of the "pingpong" sequence. Note that the curves of Figs. 7 and 8 are not strictly continuous; they are generated by connecting discrete points, corresponding to the available bit rates listed in Table I.

Table III compares the compression performances of the nonscalable MPEG-1 algorithm and the scalable low and medium delay algorithms of Table II, using all three "pingpong," "football," and "flower garden" sequences, considering chrominance as well as luminance component PSNR values. This table also indicates the form of camera motion present in each sequence. Although the scalable compression algorithms considered here are able to outperform MPEG-1 in compressing the "football" and "pingpong" sequences, the MPEG-1 algorithm is clearly superior, from the point of view of raw compression performance, in the case of the "flower garden" sequence. This is readily understood from the fact that scene motion in the "flower garden" sequence consists entirely of camera translation, which is not well approximated by a camera pan model. Nevertheless, the reader is reminded that the invertible predistortion concept introduced in Section IV-A need not be limited simply to camera pan compensation. This restricted case of global translational motion compensation is considered here only for simplicity.

D. Investigation of Constant Distortion Substream Scaling

The rate-distortion curves of Figs. 7 and 8 correspond to CBR subsets of the layered substream hierarchy. We turn our attention now to the variable bit rate (VBR) subsets generated by constant distortion substream scaling, as described in Section II. We begin by considering distortion tags, $\mathcal{D}^s_\psi$, which are generated according to $\mathcal{D}^s_\psi = \mathcal{T}(V^s_\psi)$, where the reference distortion values, V^s_ψ, are obtained from the MSE V-distortion measure, described in Section VI-A, and the fixed map, $\mathcal{T}$, is generated ahead of time for each particular video sequence, as described in Section III-A. This approach to distortion tag generation is suitable only for prerecorded video material. It is of particular interest for revealing the performance limits associated with constant distortion scaling. This is because the adaptive maps, $\mathcal{T}^s$, described in Section III-B, converge to the appropriate fixed map, $\mathcal{T}$, in the limit as the adaptation time constant, B, and the ratio s/B both tend to infinity, where s is the frame slot number. Toward the end of this section, we investigate the behavior of adaptive maps, $\mathcal{T}^s$, with finite time constants, B, and finite video sequence support.

The rate-distortion curves of Figs. 9 and 10 plot the approximately constant luminance PSNR as a function of average bit rate, $R(\mathcal{D})$, for distortion targets, $\mathcal{D} = -1, -2, \cdots, -21$.[26] For each of these mean bit rates, the instantaneous bit rate may take on any of the 28 values appearing in Table I. The figures indicate compression performance associated with the algorithms in rows one and three of Table II, these being representative examples of scalable intraframe compression and delay and memory sensitive scalable interframe compression, respectively. Source material for Fig. 9 is the 256 frame SIF525 resolution "pingpong" sequence. Fig. 10, on the other hand, indicates compression performance over a much longer video sequence of 2500 frames, taken from the movie "Raiders of the Lost Arc." This sequence is composed of three contiguous scenes, digitized from laser disc and restored to the original motion picture frame rate of 24 frames per second by discarding frames which had been duplicated during laser disc recording. The resolution in this case is 320 × 240 pixels, with chrominance components subsampled by two, both horizontally and vertically. Scene content for the "Raiders of the Lost Arc" sequence varies in activity from a wild street fight to conversational excerpts. Figs. 9 and 10 both indicate that scalable interframe compression requires approximately 0.4 to 0.6 times the average number of bits required by scalable intraframe compression, to compress the respective sequences with the same, roughly constant value of MSE distortion. This observation holds over the most interesting range of luminance PSNR values, of about 30–40 dB; the lower end of this range corresponds to noticeable, but arguably tolerable distortion, while the upper end corresponds to near visually perfect reconstruction. Convergence of the intraframe and interframe compression curves in Fig. 10, at very high PSNR, is largely due to digitization and laser disc recording noise levels, which are on the order of 40 dB PSNR.[27]

[26] These first 21 of the 28 valid distortion targets, indicated by Table I, are sufficient to reveal the most interesting region of the rate-distortion characteristic.

[27] Noise power is estimated from the MSE between digitized copies of those frames, which had been repeated in the original laser disc recording for compatibility with NTSC's 30 Hz frame rate.

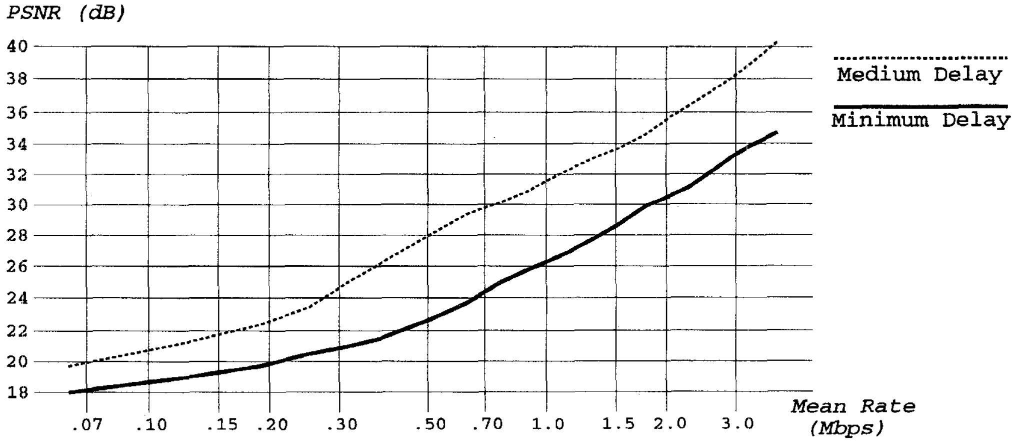


Fig. 9. Luminance PSNR of "pingpong" sequence, reconstructed after VBR substream scaling, using the first and third algorithms of Table II. Curves identified by end-to-end delay classification.

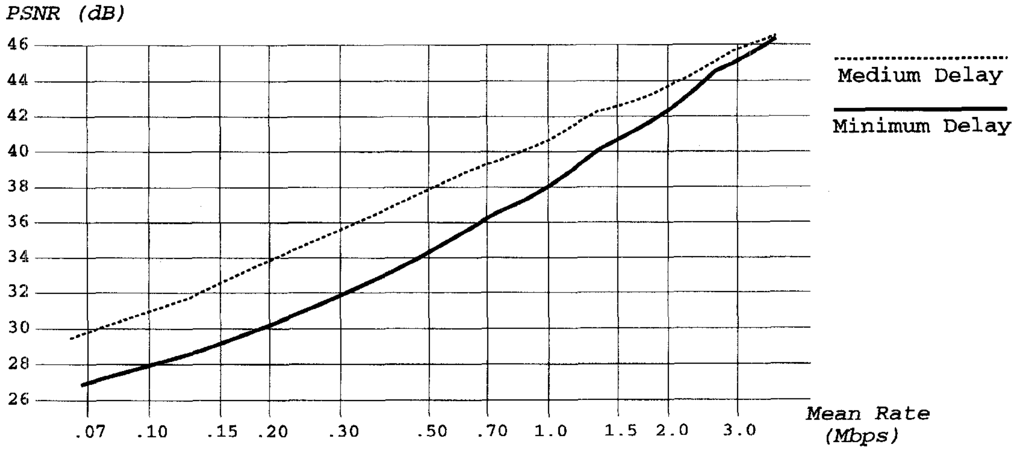


Fig. 10. Same as Fig. 9, but for "Raiders of the Lost Arc" sequence.

For further insight into the effectiveness of constant distortion scaling, Fig. 11 provides frame-by-frame luminance PSNR curves for the "pingpong" sequence, reconstructed from both CBR and VBR scaled substream hierarchies, at an average rate of $R(-14) = 1.520\,64$ Mb/s. The same scalable intraframe and interframe compression algorithms, used to generate Fig. 9, are investigated here. It is clear that the "constant" distortion VBR scaling approach significantly reduces fluctuations in the PSNR from frame to frame. Remaining variations are mainly attributable to the discretization inherent to our substream scaling approach: both instantaneous bit rates and distortion tag values belong to discrete Ψ-element sets. The overall subjective appeal associated with VBR scaling is also found to be significantly higher than that associated with CBR scaling at the same average bit rate. Subjective improvements are particularly noteworthy for interframe compression during the zooming part of the "pingpong" sequence, this motion being poorly described by our camera pan model. Fig. 12 indicates the frame-by-frame instantaneous bit rates corresponding to the VBR curves in Fig. 11. Fig. 12 clearly reveals the heightened bit rate requirements associated with interframe compression during camera zoom and scene changes. Instantaneous bit rate distributions for the much longer "Raiders of the Lost Arc" video sequence, are revealed in the histograms of Fig. 13(a) and (b). These histograms correspond to average bit rates of $R(-7) = 506.88$ kb/s for interframe compression and $R(-11) = 1013.76$ kb/s for intraframe compression, respectively, at which both algorithms give roughly similar, "low" levels of distortion. The histograms indicate that VBR scaling of a realistic video sequence can lead to widely distributed instantaneous bit rates.

As discussed in Section V, our algorithm for packaging code block quantization layers into substreams guarantees that all code bits remaining after the application of CBR scaling may be used in the decoding process. On the other hand, when constant distortion scaling is employed, it can happen that more quantization layers are available for some interframe coded code blocks in frame slot s, than were available in frame slot $s - 1$. Due to the interframe dependencies associated with generating the conditioning contexts for progressive interframe coding, these additional quantization layers cannot be decoded, in which case the associated code bits should be regarded as wasted transmission bandwidth. Of course, the compression algorithms corresponding to the first and last rows of Table II

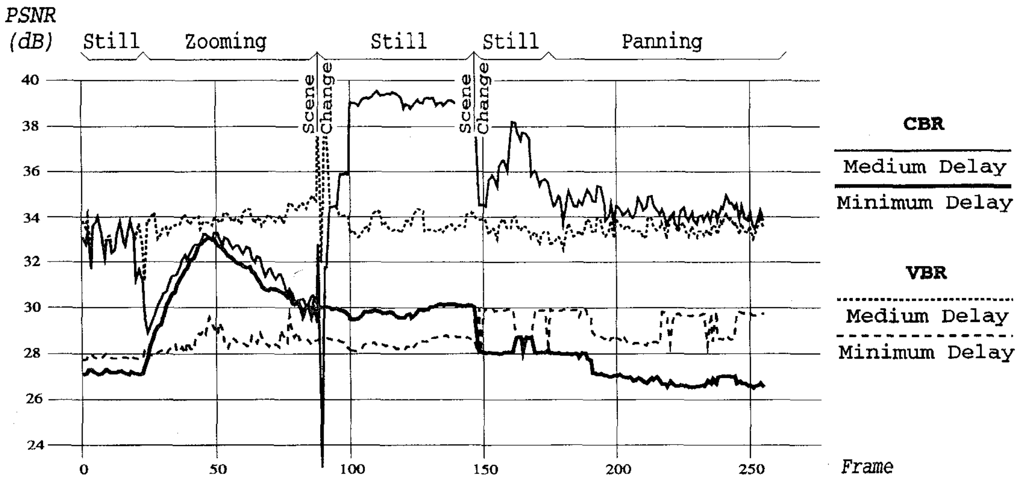


Fig. 11. Frame-by-frame luminance PSNR for "pingpong" sequence, reconstructed from CBR and VBR scaled substream hierarchies generated using the first and third algorithms of Table II. Average bit rate is 1.5 Mb/s. Curves identified by end-to-end delay classification and scaling criterion.

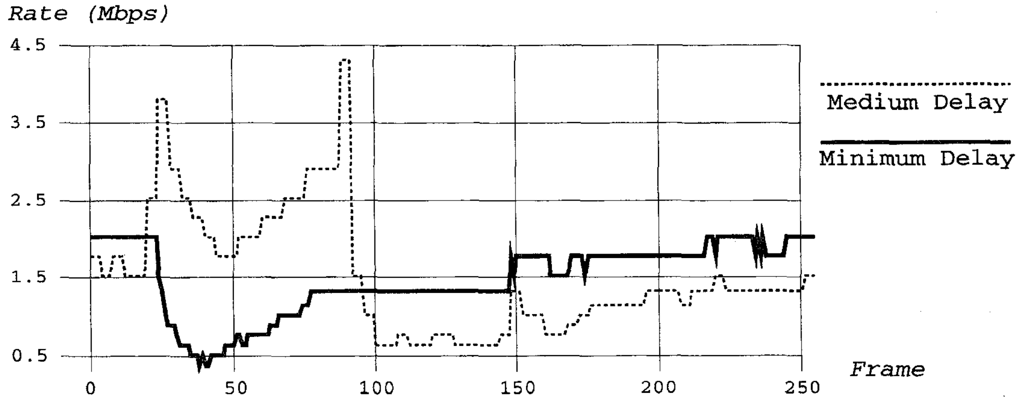
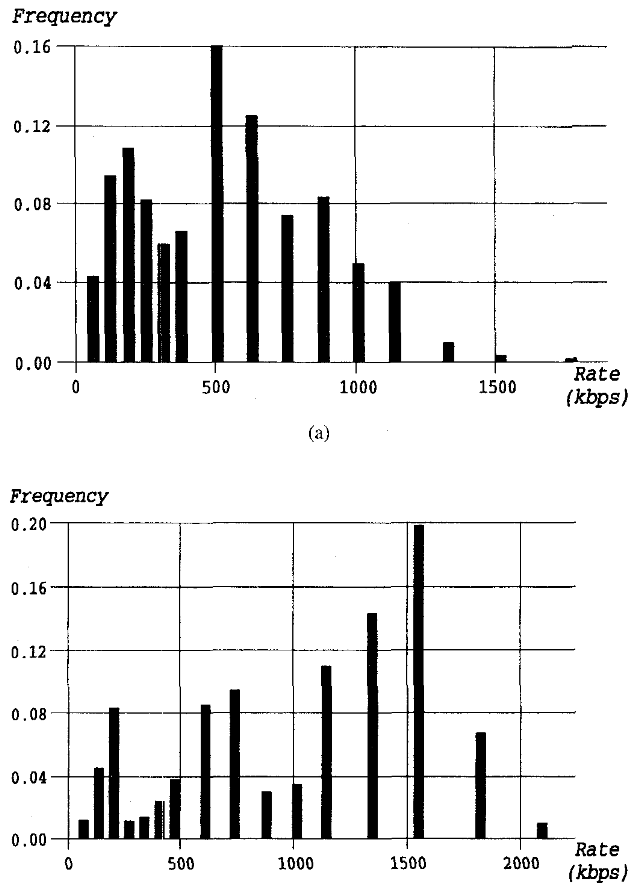


Fig. 12. Frame-by-frame instantaneous bit rates, corresponding to VBR scaling of the substream hierarchies generated from the "pingpong" sequence, using the first and third algorithms of Table II. Average bit rate is 1.5 Mb/s. Curves identified by end-to-end delay classification.

do not suffer from this bandwidth wastage problem, because the first frame of each frame slot is completely intraframe coded, i.e., $\mathcal{I} = 1$, in these algorithms. The remaining three algorithms of Table II do suffer some bandwidth wastage during constant distortion scaling. In particular, Fig. 14 reveals the overall percentage of code bits wasted by such unsatisfied interframe dependencies, when constant distortion scaling is applied to the substreams generated by the medium delay scalable compression algorithm of Table II, for both the "pingpong" and "Raiders of the Lost Arc" video sequences. The low levels of wasted transmission bandwidth indicated by Fig. 14 may be understood from the following argument. Equation (20) guarantees that unsatisfied dependencies, resulting in undecodable bits in frame slot s, may only exist provided the number of available substreams, $\psi^s(\mathcal{D})$, is greater than the number of substreams, $\psi^{s-1}(\mathcal{D})$, available in the previous frame slot. However, $\psi^s(\mathcal{D}) > \psi^{s-1}(\mathcal{D})$ suggests that more code bits are required in frame slot s than in frame slot $s-1$, in order to achieve the distortion target, $\mathcal{D}$. The event, $\psi^s(\mathcal{D}) > \psi^{s-1}(\mathcal{D})$, occurs principally because more substreams are required in frame slot s than in frame slot $s-1$ to prevent a drop in the number of code block quantization layers and hence the distortion. For this reason, the number of quantization layers allocated to any code block, β, in the first $\psi^s(\mathcal{D})$ substreams of frame slot s, may very well be no larger than the number of quantization layers allocated to code block β in the first $\psi^{s-1}(\mathcal{D})$ substreams of frame slot $s-1$, even though $\psi^s(\mathcal{D})$ is larger than $\psi^{s-1}(\mathcal{D})$. As a result, interframe dependencies are satisfied more often during constant distortion scaling than might at first be expected.

As mentioned, the results presented above are obtained using the fixed map approach of Section III-A for distortion tag generation. Before concluding this section, we offer an indication of the performance of the adaptive scheme of Section III-B. This is important because the fixed map approach is only suitable for prerecorded video material, whereas the adaptive approach is applicable both to prerecorded and interactive video. We point out, however, that we only expect the adaptive approach to be effective when the adaptation time constant, B, is relatively large and the duration of the video sequence is much larger than B. This is because the map, $\mathcal{T}^s$, must adapt slowly if constant distortion scaling is to be



(a)

(b)

Fig. 13. Instantaneous bit rate histograms, associated with VBR scaling of the substream hierarchies generated from the "Raiders of the Lost Arc" sequence. (a) Medium delay algorithm of Table II at an average rate of 0.5 Mb/s; (b) minimum delay algorithm of Table II at an average rate of 1.0 Mb/s.

meaningful; moreover the adaptive scheme only guarantees that the average bit rate over T seconds is within $B/T \times 100\%$ of its nominal value, $R(\mathcal{D})$, where $\mathcal{D}$ is the distortion target. It is difficult, therefore, to satisfactorily demonstrate the performance of the adaptive concepts in Section III-B without a large quantity of source video material. The longest video sequence available to us is the "Raiders of the Lost Arc" sequence, which has a duration of 104 s. Unfortunately, $\mathcal{T}^s$ tends to adapt too quickly when B is much less than 104 s. For illustrative purposes, therefore, we select a value of $B = 40$ s. The reference distortion bounds, $V_\psi^{\min}$ and $V_\psi^{\max}$, are obtained using the approach suggested in Section III-B, except that the "Raiders of the Lost Arc" sequence itself is used as training material for want of a more realistic training set. That is, $V_\psi^{\min}$ is selected so that $V_\psi^s \geq V_\psi^{\min}$ for 99% of the frame slots, s. Similarly, $V_\psi^s \leq V_\psi^{\max}$ for 99% of the frame slots. This ensures that the reference distortion values, V_ψ^s, must occasionally be artificially constrained to lie within the bounds $V_\psi^{\min} \leq V_\psi^s \leq V_\psi^{\max}$, as we would expect if the bounds were generated from a realistic set of training video sequences. Also, the initial fidelity threshold values a_i^1 are set to $a_i^1 = [\mathcal{M}(V_{-i}^{\max}) + \mathcal{M}(V_{-i}^{\min})]/2$, where $\mathcal{M}$ is the piecewise linear map described in Section III-B.[28]

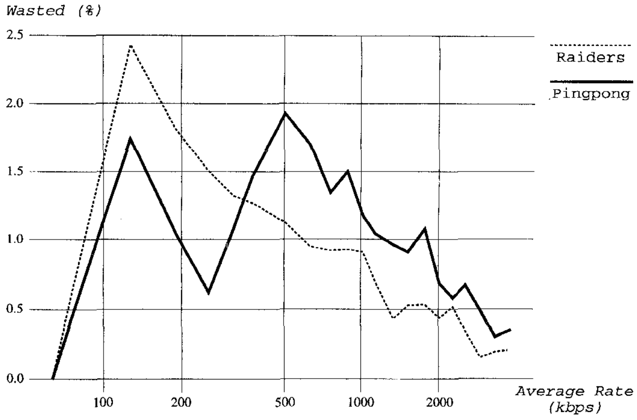


Fig. 14. Percentage of bits wasted due to VBR scaling of the substream hierarchies generated from the "pingpong" and "Raiders of the Lost Arc" sequences, using the medium delay algorithm of Table II.

Fig. 15 indicates the frame-by-frame PSNR values obtained for the "Raiders of the Lost Arc" sequence, after both constant bit rate scaling and constant distortion scaling, with an adaptation time constant of $B = 40$ s. The scalable data stream is obtained using the medium delay algorithm of Table II and the nominal average bit rate is $R(-7) = 506.88$ kb/s. As seen, constant distortion scaling does offer some smoothing of the distortion, especially in the first 1000 frames, where constant bit rate scaling exhibits the widest fluctuations in distortion. However, the map clearly adapts too quickly to offer good long term stabilization of the PSNR or, equivalently, MSE. The distortion in Fig. 15 is seen to wander appreciably, e.g., 1–2 dB, within a period of about 4 s, suggesting that the time constant, B, required to hold distortion approximately constant over a period of one minute, for example, may need to be as large as 10 min.

As discussed in Section III-B, $\mathcal{T}^s$ must be adapted in such a way as to satisfy the leaky bucket criterion of (7). Fig. 16 plots the normalized bucket fullness ratio

$$\frac{\left\{\left[\sum_{s=1}^{S}\sum_{\psi=1}^{\psi^s(\mathcal{D})} R_\psi\right] - SR(\mathcal{D})\right\}\cdot\mathcal{F}}{F_R R(\mathcal{D}) B}$$

over the range $1 \leq S \leq 2500/\mathcal{F}$, for various distortion targets, $\mathcal{D} = -4, -7, -11$, and -16, corresponding to nominal average bit rates of 253.44 kb/s, 506.88 kb/s, 1013.76 kb/s, and 2027.52 kb/s, as indicated in Table I. According to (7), the absolute value of this ratio should always be less than or equal to one. As seen from Fig. 16, this is indeed the case. Moreover, Fig. 16 indicates that the bounds of ±1 are equally tight over a wide range of distortion targets.

As mentioned in Section III-B, it can happen that the reference distortion values must occasionally be modified in order to avoid misordering of the fidelity threshold values,

[28] Recall from Section VI-A that $\psi_i^a = \psi_i^b = -i$ for each distortion target, $d_i = -i$.

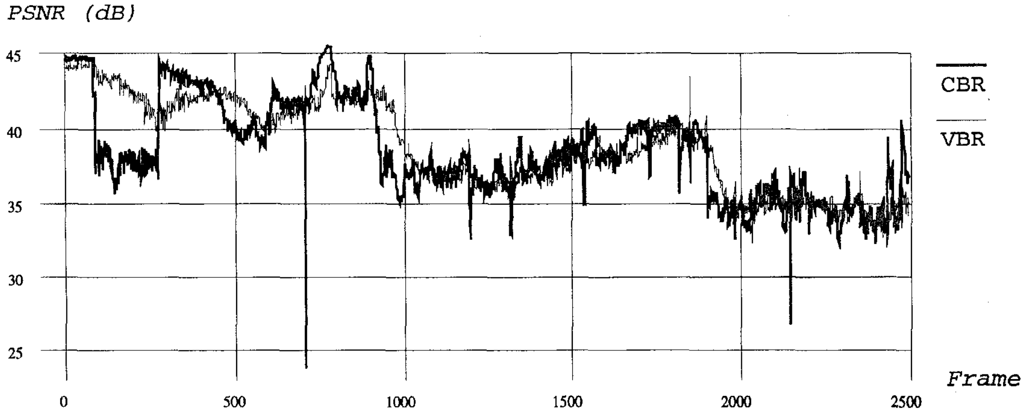


Fig. 15. Frame-by-frame luminance PSNR for "Raiders of the Lost Arc" sequence, reconstructed from CBR and VBR scaled substream hierarchies generated using the medium delay algorithm of Table II. Nominal average bit rate is 506.88 kb/s. Adaptation time constant for adaptive distortion tag map, $\mathcal{T}^s$, is $B = 40$ s.

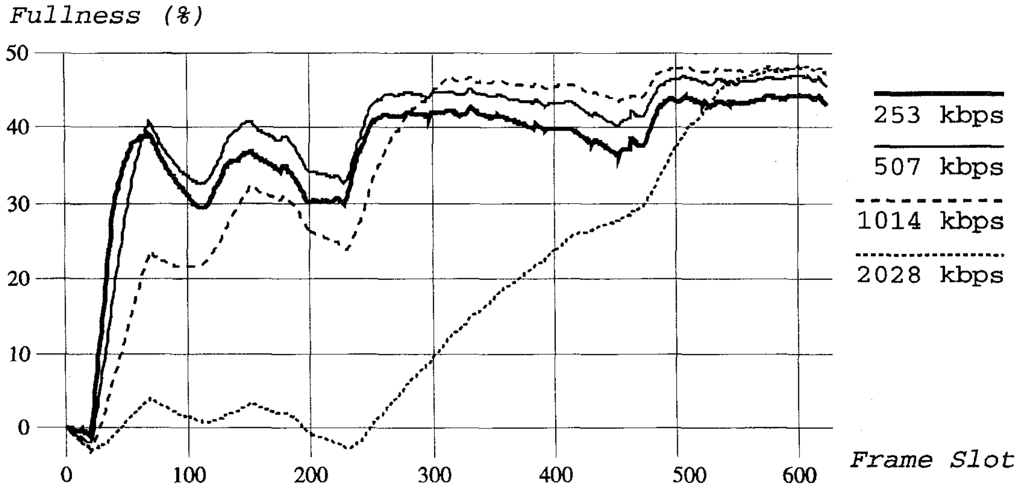


Fig. 16. Normalized bucket fullness ratio, corresponding to the first S frame slots of the constant distortion scaled substream hierarchy generated using the medium delay algorithm of Table II. Adaptation time constant for adaptive distortion tag map, $\mathcal{T}^s$, is $B = 40$ s.

$a_1^s < a_2^s < \cdots < a_p^s$. The fact that this modification is required only rarely is confirmed in our experimental investigations. In particular, Fig. 17 indicates the percentage of frame slots in which one or more reference distortion values must be modified in order to prevent threshold misordering, as a function of the time constant, B. As predicted in Section III-B, the number of modifications decreases rapidly as B increases. In fact, no modifications whatsoever are required for $B > 18$ s. Given that B may well be on the order of about 10 min in a practical system, Fig. 17 suggests that the need to modify reference distortion values may virtually never arise.

VII. Conclusion

The principle contribution of this paper is the introduction of a layered substream abstraction to facilitate simple, generic scaling of highly scalable compressed data, with both constant bit rate and constant distortion (VBR) scaling criteria. The behavior of these scaling policies has been experimentally demonstrated within the context of a class of highly scalable video compression schemes, which permit compression performance to be traded for delay and/or implementation memory requirements. The conclusion of these experiments is that compression performance similar to that of MPEG-1 should be attainable with similar end-to-end delay, while offering a high degree of scalability using simple, generic scaling mechanisms. The proposed layered substream hierarchies provide a particularly useful tool for rate scaling within high speed, shared digital networks, where computational resources are often relatively limited. Moreover, the generic nature of the associated scaling mechanisms renders the proposed layered substream abstraction suitable for distribution of a wide variety of highly scalable data streams. Exactly the same substream abstraction should, for example, be equally appropriate for highly scalable compressed audio data streams, which might be generated using similar approaches to those described here for video. Interesting application possibilities emerge as a result of the potential for both constant rate and constant distortion based scaling. For example, one might envisage a heterogeneous multicast environment, in which compressed video is delivered to some clients at constant bit rate, while to others with constant distortion, depending on the capabilities of their respective distribution paths. It should be noted that our work on constant distortion substream scaling has focused on

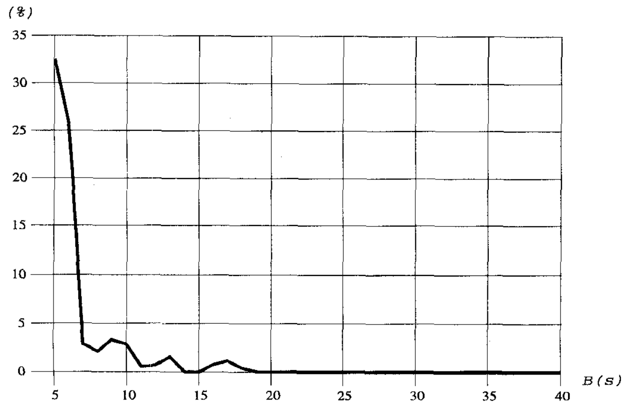


Fig. 17. Percentage of frame slots requiring slight modification to the reference distortion values, in order to prevent misordering of fidelity thresholds during adaptive distortion tagging, as a function of the time constant, B. Obtained using the medium delay compression algorithm of Table II and the "Raiders of the Lost Arc" video sequence.

the MSE distortion measure for demonstrative purposes only. More realistic psychovisual distortion measures must clearly be investigated for practical applications.

Appendix

In this appendix, we prove Observations 1 and 3. The proof of Observation 2 is essentially identical to that of Observation 1.

Observation 1:

Proof: Note that $\mathcal{D}^s_{\psi^s(d_i)} = d_j$ for some j. According to (1), we must have $d_j \leq d_i$, i.e., $j \geq i$. From (4) then, $V^s_{\psi^s(d_i)} \leq t_j \leq t_i$, so that $\psi^s(d_i) \geq \min\{\psi | V^s_\psi \leq t_i\}$. On the other hand, suppose that $V^s_\psi \leq t_i$, for some ψ. Then $\mathcal{D}^s_\psi \leq d_i$, by (4) which implies that $\psi^s(d_i) \leq \psi$, by (1). We conclude that $\psi^s(d_i) \leq \min\{\psi | V^s_\psi \leq t_i\}$. ■

Observation 3:

Proof: Consider $a_i^{\max}$. By assumption, the initial value $a_i^1 < \Phi^{\max}_{\psi_i^a} + [R(d_i) - R_1]$. Suppose also that $a_i^s < \Phi^{\max}_{\psi_i^a} + [R(d_i) - R_1]$ for some frame slot s. We have two cases.

1) Case 1 ($a_i^s \geq \Phi^{\max}_{\psi_i^a}$)

   In this case we have

$$\begin{aligned}\psi^s(d_i) &= \min\{\psi | \Phi^s_\psi \geq a_i^s\} \\ &\geq \min\{\psi | \Phi^s_\psi \geq \Phi^{\max}_{\psi_i^a}\} \geq \psi_i^a,\end{aligned}$$

   where we have used (9) and the fact that the maximum fidelity, $\Phi^{\max}_\psi$, associated with the first ψ substreams, is a nondecreasing function of ψ. It follows that

$$\sum_{\xi=1}^{\psi^s(d_i)} R_\xi - R(d_i) \geq \sum_{\xi=1}^{\psi_i^a} R_\xi - R(d_i) \geq 0$$

   and so, according to (10), we have $a_i^{s+1} \leq a_i^s < \Phi^{\max}_{\psi_i^a} + [R(d_i) - R_1]$.

2) Case 2 ($a_i^s < \Phi^{\max}_{\psi_i^a}$)

   Observe that $\sum_{\xi=1}^{\psi^s(d_i)} R_\xi - R(d_i) \geq R_1 - R(d_i)$ so that, according to (10), we have $a_i^{s+1} \leq a_i^s + [R(d_i) - R_1] < \Phi^{\max}_{\psi_i^a} + [R(d_i) - R_1]$.

By induction on s, we see that $a_i^s < \Phi^{\max}_{\psi_i^a} + [R(d_i) - R_1]$ for all frame slots $s \geq 1$. A similar argument establishes the lower bound, $a_i^{\min}$. ■

References

[1] E. H. Adelson, E. Simoncelli, and R. Hingorani, "Orthogonal pyramid transforms for image coding," in *Proc. SPIE,* vol. 845. Cambridge, MA, Oct. 1987, pp. 50–58.

[2] F. Bosveld, R. Lagendijk, and J. Biemond, "A refinement system for hierarchical video coding," in *SPIE Symp. Visual Communications and Image Processing,* vol. 1360, pt. 1. Lausanne, Oct. 1990, pp. 575–586.

[3] N. Chaddha, M. Vishwanath, and P. Chou, "Hierarchical vector quantization of perceptually weighted block transforms," in *5th Data Compression Conf.,* Snowbird, UT, Mar. 1995, pp. 3–12.

[4] E. Chang and A. Zakhor, "Scalable video data placement on parallel disk arrays," in *IS&T/SPIE Symp. Electronic Imaging, Science and Technology,* vol. 2185. San Jose, Feb. 1994, pp. 208–221.

[5] T. Chiang and D. Anastassiou, "Hierarchical coding of digital television," *IEEE Comm. Mag.,* vol. 32, pp. 38–45, May 1994.

[6] M. R. Civanlar and A. Puri, "Scalable video coding in frequency domain," in *SPIE Symp. Visual Comm. and Image Processing,* vol. 1818, pt. 3. Boston, Nov. 1992, pp. 1124–1134.

[7] W. Equitz and T. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory,* vol. 37, no. 2, Mar. 1991.

[8] A. Erdem and M. Sezan, "Scalable extension of MPEG-2 for coding 10-bit video," in *SPIE Symp. Image and Video Compression,* vol. 2186. San Jose, Feb. 1994, pp. 245–256.

[9] Z. Gao, F. Chen, B. Belzer, and J. Villasenor, "A comparison of the Z, $E8$ and Leech lattices for image subband quantization," in *Proc. 5th Data Compression Conf.,* Snowbird, UT, Mar. 1995, pp. 312–321.

[10] B. Girod, "Scalable video for multimedia workstations," *Comput. and Graphics,* vol. 17, no. 3, pp. 269–276, May/June 1993.

[11] C. Guillemot and R. Ansari, "Layered coding schemes for video transmission on ATM networks," *J. Visual Comm. Image Representation,* vol. 5, no. 1, pp. 62–74, Mar. 1994.

[12] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 11, no. 7, pp. 674–693, July 1989.

[13] G. Morrison and I. Park, "COSMIC: A compatible scheme for moving image coding," *Signal Proc.: Image Commun.,* vol. 5, pp. 91–103, Feb. 1993.

[14] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 4, pp. 339–356, June 1994.

[15] J. Ohm, "Advanced packet-video coding based on layered VQ and SBC techniques," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 3, no. 3, pp. 208–221, June 1993.

[16] A. Reibman and B. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 2, no. 4, Dec. 1992.

[17] A. Said and W. Pearlman, "Image compression using the spatial-orientation tree," in *Int. Symp. Circuits and Systems,* vol. 1. Chicago, May 1993, pp. 279–282.

[18] J. M. Shapiro, "An embedded hierarchical image coder using zerotrees of wavelet coefficients," in *3rd Data Compression Conf.,* Snowbird, UT, 1993, pp. 214–223.

[19] A. Singh, J. Bove, and V. Mkhael, "Multidimensional quantizers for scalable video compression," *IEEE J. Select. Areas Commun.,* vol. 11, no. 1, pp. 36–45.

[20] M. J. T. Smith and S. L. Eddins, "Analysis-synthesis techniques for subband image coding," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, pp. 1446–1456, Aug. 1990.

[21] J. Rissanen and G. Langdon, "Arithmetic coding," *IBM J. Research Develop.,* vol. 23, no. 2, pp. 149–162, Mar. 1979.

[22] G. Sullivan, "Optimal entropy constrained scalar quantization for exponential and Laplacian random variables," in *Proc. ICASSP,* vol. 5. Adelaide, 1994, pp. 265–268.

[23] D. Taubman and A. Zakhor, "Orientation adaptive subband coding of images," *IEEE Trans. Image Processing,* vol. 3, no. 4, pp. 421–437, July 1994.

[24] ——, "Multi-rate 3-D subband coding of video," *IEEE Trans. Image Processing,* Special Issue on Image Sequence Compression, vol. 3, no. 5, pp. 572–588, Sept. 1994.

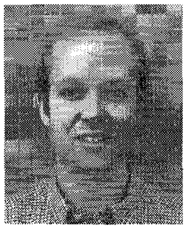
[25] ——, "Highly scalable, low-delay video compression," in *Proc. 1st Int. Conf. Image Processing,* vol. 1. Austin, Nov. 1994, pp. 740–744.

[26] D. Taubman, "Directionality and scalability in image and video compression," Ph.D. dissertation, University of California, Berkeley, 1994.

[27] P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[28] L. Vandendorpe, "Hierarchical transform and subband coding of video signals," *Signal Processing: Image Commun.,* vol. 4, pp. 245–262, June 1992.

[29] J. Woods Ed., *Subband Image Coding.* Norwell, MA: Kluwer, 1991.



David Taubman (S'93–M'95) 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. and Ph.D. degrees from the University of California at Berkeley in 1992, and 1994, respectively.

From 1988 to 1990, he worked for the Electricity Commission of NSW, Australia. In 1994, he joined Hewlett-Packard Laboratories in Palo Alto, CA, where he is currently involved in image and video enhancement research. His research interests include inverse problems for image and video applications, image and video compression, and the interaction between compression and networking issues.

Dr. Taubman was awarded the Sydney University Medal for Electrical Engineering in 1988. In the same year, he also received the Institute of Engineers, Australia, Prize and the Texas Instruments Prize for Digital Signal Processing, Australia. He was a University of California Regents Fellow from 1990 to 1991.

Avideh Zakhor (S'87–M'87) received the B.S. degree from the California Institute of Technology, Pasadena, and the S.M. and Ph.D. degrees from the 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 at 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, holds four U.S. patents, and is the co-author of the book, *Oversampled A/D Converters* with Soren Hein.

Dr. 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 a member of the technical committee for image and multidimensional digital signal processing.