Multi-Resolution Layered Coding for Real-Time Image Transmission: Architectural and Error Control Considerations

James K. Han and George C. Polyzos

Computer Systems Laboratory
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114, U.S.A.

Abstract— **We examine the rationale for and systems aspects of using layered coding for the transmission of images and video over heterogeneous communication networks. We advocate Multi-resolution Layered Coding (MLC) and present architectural alternatives that can support it within the scope of existing standards and technologies. Since the premise of layered coding is controlled loss, the issues of error control and concealment are central. We demonstrate that MLC exhibits good error tolerance, effectively compensating for errors in situations where traditional error concealment schemes, based on the interpolation of pixels in adjacent blocks, are ineffective. In particular, MLC yields a higher Peak Signal-to-Noise Ratio (PSNR) than two-layer coding when packet loss occurs in the base layer. Finally, we show that MLC can effectively exploit efficient network error control strategies which distribute redundancy non-uniformly and provide unequal levels of protection across layers.**

I. Introduction

The role of images in human communication is central. Because of the lack of transmission bandwidth and system support for them, images were all but excluded from mainstream computing and data communications until very recently. In the last few years however, digital image and video have been a driving force for the design and deployment of high-speed communication networks and distributed multimedia applications. Furthermore, the direction is towards full integration of transmission and handling of the various media, with a Broadband Integrated Services Digital Network (B-ISDN) one of the best examples of the goals to be achieved.

The success of the World Wide Web, following the availability of point-and-click browsers and improvements in the image and multimedia capabilities of browsers and distributed multimedia applications, provides additional incentives for the full integration of digital images and video in everyday computing. It has also fueled increasing pressure for user customization of media presentation.

Note that differentiation in data handling based on (media) content is important mainly when real-time presentation (or processing) of the media is required. Otherwise, images and video can (and probably should) be treated as any other data transported by communication networks or processed by computer systems.

Address correspondence to G. C. Polyzos at the above address, or contact him at Tel.: +1–619–534–3508, FAX: +1–619–534–7029, Email: polyzos@cs.ucsd.edu.

A. Interactive Multimedia over Packet Networks

Due to the very high data volumes of images, audio, and video it is important to provide efficient network mechanisms for these types of traffic even when high-speed fiber-optic networks are available, but more so if heterogeneous networks with possibly severely bandwidth constrained links or subnets (e.g., wireless) are considered. In addition to very high throughput requirements, images, audio and video are usually associated with real-time interactive applications and thus they also present rather stringent delay constraints. More precisely, in order for audio and video to be effectively used in interactive communication, i.e., without forcing the communicating subjects to modify their behavior from that of face-to-face communication, the delay between transmitter and receiver is expected to be minimal. Various guidelines set this end-to-end delay tolerance in the tenths or few hundreds of ms [1].

Traditionally, transmission of Continuous Media (CM), such as real-time audio and video, has been based on Constant Bit Rate (CBR) circuit-switching channels. Therefore, source coding schemes for CM were designed specifically to produce output at a specific, constant, bit rate, typically that of the channel to be used, independently of the instantaneous information content of the signal to be transmitted. This coding typically results in variable signal quality, even though when bandwidth is not severely limited, potential temporary quality degradation is usually engineered to be imperceptible.

For transmission over packet-switching networks, the problem of CM coding changes. There is typically no a priori constraint on the maximum instantaneous bit rate produced by the coding scheme, but instead, an effort is made to produce constant quality signal at a given average bit rate (or level of quality) by using Variable Bit Rate (VBR) schemes. Peak-to-mean bit-rate ratios for VBR codecs is usually high; a 4.7 value is reported for one particular codec [2].

The use of VBR coding is based on the capability of packet-switching networks to use statistical multiplexing for efficient utilization of network resources. One can then economize on resources when the information content of the signal is low and expend them at a higher rate when it is high, achieving the best overall (constant) quality at a given cost. The potential for significant economies from statistical multiplexing was one of the main arguments for the selection of the Asynchronous Transfer Mode (ATM), the transmission and multiplexing mode standardized for B-ISDN, over the circuit-switching alternative.

Of course, there will be instances where peak demands are presented to the system (actually, its "best-effort" component) simultaneously. In most of these cases the system will be unable to immediately satisfy these demands. In traditional packet-switching networks these situations have been handled through queueing of the requests and congestion control. With real-time services, however, queueing might introduce unacceptable delays. Therefore, one of the techniques considered for traffic smoothing and controlling user perceived latency in times of high congestion is the dropping of parts of the signal that might be of secondary importance. This approach is made possible by the use of hierarchical or layered coding and appropriate traffic labeling and prioritization.

B. Layered Coding

Hierarchical or Layered Coding (LC) techniques split signals into components of varying importance [3], [4]. The aggregation of these components reconstructs the original data, but subsets of the data can also provide various degrees of approximation to the original signal. Signal subsets are coded separately (possibly with different coding schemes), and can therefore be decoded separately. By careful design, the first (or first few) components in the hierarchy can be a good approximation to the overall signal, providing a good first impression of the information without requiring all the components to be received (and decoded) first. LC has advantages in many facets of image and video transmission [5], [6], [7].

LC with a variable spatial resolution [8], has received less attention than LC with a fixed resolution, despite its adoption in some popular coding standards [9], [10], [11]. Two-layer coding has been mostly considered in the literature [2], [5], [7], [12], [13], with the first, coarse or low frequency, subsignal usually referred to as the *base layer*, and the second, higher resolution, subsignal termed the *enhancement layer*.

We use the term Multi-resolution Layered Coding (MLC) to refer to LC with more than two layers of spatial resolution. One well-known form of MLC is pyramidal coding, with variable spatial resolution across layers [8]. Our intention here is to differentiate MLC from LC with a spatial fixed resolution hierarchy (e.g., subband coding) or even from single resolution LC using partitioning of the (transform space) coefficients into two or more layers for transmission (e.g., see [15]). We focus on MLC because schemes with variable spatial resolution provide better image quality at multiple and lower resolution levels[9](p. 96). This is a very desirable feature for some important applications which require good quality using the lower layers or tight control over the coding error [14]. Actually, with MLC the quality can be optimal at the specific resolutions selected to define the layers in the hierarchy. The main potential disadvantage of LC, and MLC in particular, is the increased overhead; this is discussed in section 2.

C. Hierarchical JPEG

The existing JPEG image compression standard has a provision for MLC, referred to as the hierarchical mode [9] of JPEG (HJPEG). This mode of JPEG is based on pyramidal coding and has different spatial resolution at each layer.[1] A block diagram of the codec is shown in Fig. 1. The image is first low pass filtered and subsampled by the desired number of multiples of 2 in either or both dimensions and encoded using either the *sequential* or the *progressive* JPEG mode [9]. Then the encoded reduced-size image is decoded, interpolated and upsampled by the same number of multiples of 2 horizontally and/or vertically. This upsampled image is then used as a prediction of the original image at this resolution and the difference image is computed, encoded, and transmitted. Finally, the last two steps are repeated until the original image at full resolution has been encoded.

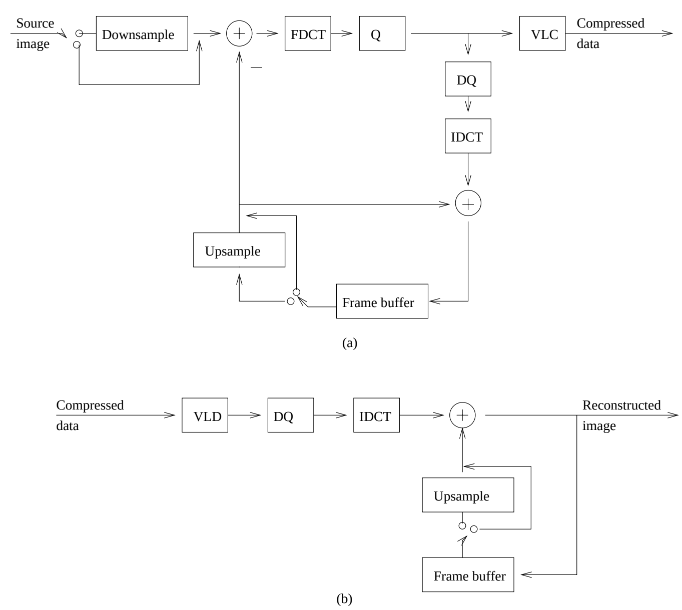


Fig. 1. Block diagram of the Hierarchical JPEG (HJPEG) (a) encoder and (b) decoder. FDCT: forward discrete cosine transform, Q: quantization, VLC: variable length (Huffman/arithmetic and run-length) coding, IDCT: inverse discrete cosine, DQ: dequantization, VLD: variable length decoding

We have implemented in software the HJPEG codec [10] and have used it to obtain the experimental results we present in sections 2 and 4 of this paper, investigating the properties of LC when used for image and video transport.

[1] Layers are called "frames" in JPEG terminology, but we prefer to use the term "layer" here to avoid confusion with video frames.

Note that even though JPEG has been designed for still images, it is also widely used for video transmission, providing (only) intraframe compression. Furthermore, because this scheme uses only intraframe compression it is much more robust in the presence of errors than coding methods exploiting both temporal and spatial redundancy, such as MPEG.[2]

D. Overview and Contributions of this Paper

The remainder of this paper is structured as follows. Section 2 presents potential applications of LC and their implications for network services, as well as the disadvantages of using these techniques. Section 3 discusses architectural alternatives for various packet network technologies that can effectively support and exploit the features of Layered Coding (LC), and Multi-resolution Layered Coding (MLC) in particular. Section 4 deals with the impact and handling of transmission errors, focusing on concealment techniques and the implications for LC and network architectures. Finally, we present our conclusions in section V.

The main contributions of this paper are the following. First, the proposal and rationale for using MLC for image and continuous media transmission over packet networks, especially in order to support multicast and heterogeneity (particularly, for both high-speed fiber-optic networks and wireless networks, with high error-rates). Second, the proposal to use multiple ATM virtual-channels with different Quality of Service (QoS) levels instead of the standard two-level priority scheme based on the Cell Loss Priority bit. Third, the proposal to use both guaranteed QoS connections for the critical, but highly compressible parts of CM signals, and "best-effort" connections for the enhancement layers. And finally, our analysis of the impact of transmission errors, leading to cell or packet loss, on MLC schemes.

II. Rationale for Layered Coding

A. Layered Coding Features Improving Application Responsiveness

The following are some potential applications of Layered Coding (LC) for image and video transmission:

1. *Progressive presentation:* LC can be exploited in order to reduce the *perceived* transmission and presentation latency for image database browsing. An image can be painted progressively on the screen, painting a low resolution version of the whole image first (with low delay because of the much smaller size of the base layer). Detail can be added progressively, by adding subsequent layers as they arrive and are decoded, increasing the resolution to the desired level or up to the maximum available. This feature is important for applications that involve repeated, continuous display of images with inspection by a human, i.e., browsing an image database until a desired image is found. The real value is obtained by having the user abort the transmission early, immediately after realizing that an image shown is not the desired one. This determination can usually be made much sooner than the end of the transmission, decoding, and display process for a complete non-LC image, or for all the layers in a LC image, depending of course on image size and resolution, the parameters of communication bandwidth and latency, and the end-system reconstruction and display latency.
2. *Multiresolution capability:* High resolution images can easily be used by low resolution devices through LC. This is particularly relevant for small portable terminals and for conferencing where multiple smaller images need to share the screen with full size images at different times.
3. *Reduced bandwidth transmission:* High quality images can be transmitted faster over low-bandwidth channels, at lower resolution.
4. *Resolution refinement:* This is the ability to produce still images at higher resolution than when viewed in motion [4]. If the various frames in a motion video are hierarchically coded, then only the acceptable resolution components can be decoded at the receiver, while the higher resolution components can be cached, to be used only for still frame display. This reduces the decoding time for the common case.
5. *Zooming:* This is the capability of scaling an image up or down. This can be accomplished by switching between low-resolution downscaled picture icons and the full-scale picture at high-resolution.
6. *Selective resolution reduction:* Images can be compressed with different resolutions across areas of the image by taking into consideration the importance of particular portions of the picture in order to produce a more compact image.
7. *Security/Privacy:* LC can be used to allow only some of the users to view the images at full resolution, while others can only view low-resolution versions.

We are mainly interested in features 1–3, which can be used to improve efficiency in packet-networks and we show how in section II-C. However, we first discuss some potential difficulties with LC.

B. Disadvantages of Layered Coding

Two are the main problems with LC: increased overhead (lower compression ratios) and increased processing complexity. We discuss and analyze them next and also present other potential disadvantages or issues that need to be addressed.

B.1 HJPEG Compression Measurements

Layered coding is known to provide lower compression ratios than non LC at the same image quality. Viewed from a different angle, LC leads to increased overhead. A rough idea of the amount of overhead introduced, assuming downsampling by 2 in each dimension for each layer, can be obtained by considering the *relative* size of the layers, starting with the top, full resolution, layer: 1, 1/4,

[2]MPEG uses intraframe coded only frames periodically to achieve a desired level of redundancy; at the extreme, intraframe coded frames could be used exclusively, obtaining similar levels of robustness and compression with motion JPEG.

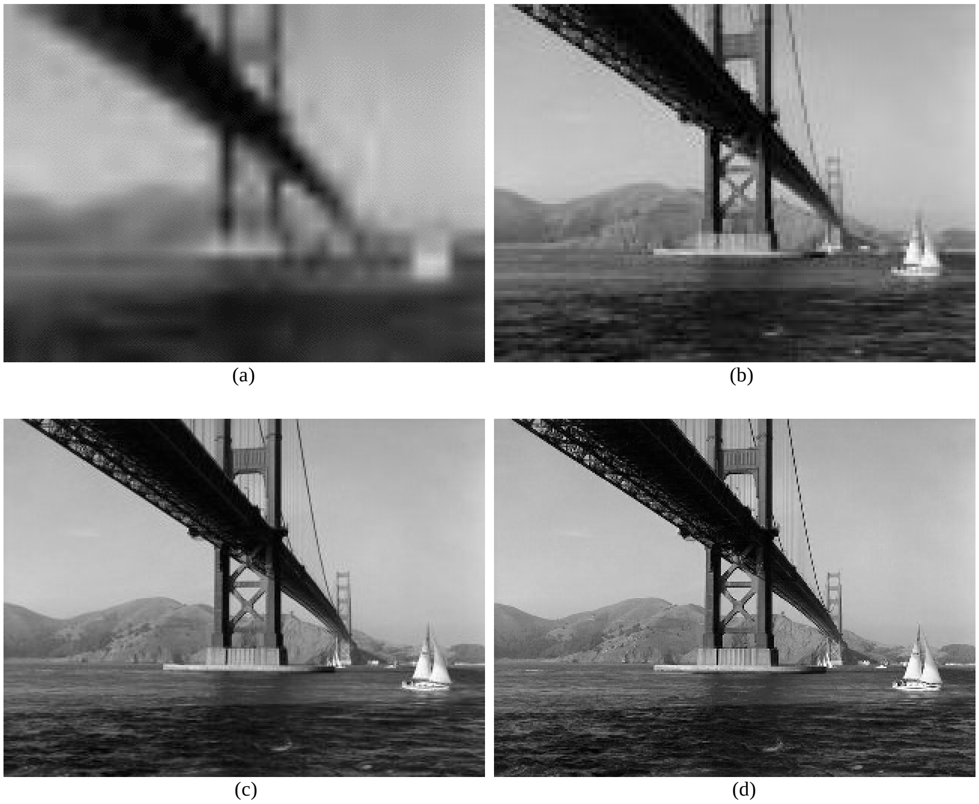


Fig. 2. Hierarchical progression up to the (a) first (80x60, 0.048 bpp), (b) second (160x120, 0.093 bpp), (c) third (320x240, 0.291 bpp), and (d) fourth layer (640x480, 1.029 bpp) of 4-layer HJPEG image (bridge). Images (a)-(c) have been expanded to the size of the original image (640x480) for comparison.

$(1/4)^2, \ldots$ The sum of the infinite series is 1.33. I.e., there is 25% overhead for two layers and up to 33% for any number of layers. These figures should be considered "upper bounds" because they essentially assume that the second and subsequent layers are used in addition to the full-size image. On the other hand, this analysis does not account for possible repetition of tables and other header information included in coded images.[3]

To get a better understanding of the overhead of LC, we experimented with two sets of image data. The first data set consists of various still images from the public domain distribution of JPEG images. These are completely independent, self contained images. The second data set was obtained from a 5-second video clip (149 frames) of a football game. In this case the images are inherently correlated. In the first data set, the overhead of the HJPEG files tends, as the size of the images increases, to converge to about 13%, 18%, and 20% for 2-layer, 3-layer, and 4-layer HJPEG images, respectively. These compression ratios may vary slightly across HJPEG implementations (influenced by different upsamplers and downsamplers), but show far better performance than expected, considering the initial results,[4] reported in [9]. Note also that compression ratio also depends on image content and quantization level.

We have observed that downsampling of reasonable size images (from 200×150 and up) by 2 both horizontally and vertically yielded almost as acceptable image quality as the full-resolution image (Fig. 2). Note that the total size, up to and including the third layer of the 4-layer HJPEG image, constitutes only 35% of the Baseline JPEG (BJPEG) size and takes 43% of the time to decode compared to the BJPEG. In the case of 2-layer HJPEG compression of the same image, these figures are 29% and 37%, respectively. We believe that the results for the compression ratio are more important than those for decoding time because their range of improvement is limited, while the processing time can be significantly reduced through optimization of the software or implementation in hardware.

We have performed the same analysis to the second data set, but we report summary results in the form of mean and standard deviation of the sizes across the set, rather than numbers for individual images. The mean size for 2-layer HJPEG coding increases by 10.8% (over BJPEG) with a standard deviation σ of 0.78% for 2 layers. The corresponding numbers for 3 and 4 layers are 14.7% with a

[3] An additional source of increased overhead in HJPEG is the process of upsampling and downsampling used within the intraframe compression technique. An inefficient sampling filter can cause further error corrections in the following layers and lead to a lower compression ratio.

[4] In [9] comparison was made between a 3-layer HJPEG image (downsampled by 2 in each direction for each layer) and the progressive mode of JPEG. The HJPEG image has about 33% overhead over the image coded with the progressive mode.

σ of 1.1% and 16.8% with a σ of 1.3%, respectively. These figures show a rather higher compression ratio than the previous data set consisting of still images. This is mainly due to the constant signal intensity of low spatial frequency from the plain lawn ground in the scene. Note that scene changes have no effect on the bit rate because the temporal redundancy of the images is not exploited.

Another observation is that the file size of the progression of the layers up to the $(n-1)$-st layer (frame in JPEG terminology) does not depend strongly on the number of layers (n) and is approximately 25-30% of the BJPEG file. This is compatible with the above asymptotic analysis. Note that although the file size and the decompression time increase slightly as the total number of layers used in the HC scheme (i.e., n) increase (e.g., see Fig. 3, and 4), the image quality achieved by using the first $n-1$ layers is almost invariant in a subjective evaluation. In this case, the larger prediction errors in the first $n-2$ differential layers of HJPEG with more layers contribute to a slightly larger size (Fig. 3) without leading to a better perceptual quality than that of HJPEG with fewer layers. This suggests that the number of layers should be decided according to the priority scheme supported by the underlying networks, if finer granularity control of the image resolution is not desired. The impact of network errors on multiple layer images is discussed in section IV-C.1.

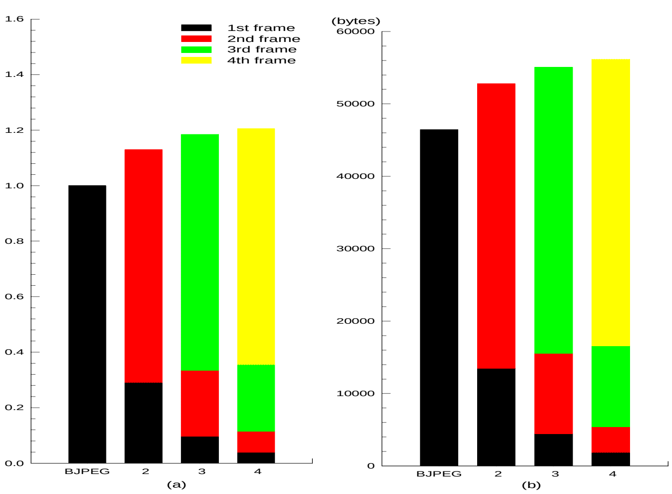


Fig. 3. (a) Ratio of HJPEG layer sizes to BJPEG size; (b) size of layers in an n-layer HJPEG image (bridge).

B.2 Synchronization

Splitting signals into layers and transmitting the layers separately through the network introduces the need for synchronizing the various layer streams before they can be displayed at the receiver. However, even though stream synchronization seems to be a new requirement added by the use of LC, in reality it is just an aggravated version of the inter-media (e.g., audio-video) synchronization problem, which is central but solvable for real-time continuous media.

Note that the traditional solution to the inter-media synchronization problem based on bundling together all the media streams into a single "wide" stream (for example, by appending the corresponding audio segment to a video frame [16]), even though it greatly simplifies the problem, it misses any opportunity for medium specific treatment within the network and at the end-terminal. For example, sudden congestion at a network point would lead to complete disruption of service, i.e., freezing of the video and audio. However, if priorities are supported by the network, because of the importance of audio and its usually much smaller bandwidth requirements, a better solution would be to continue the audio service and discontinue temporarily the video. If the media are bundled together, there is no way for a network switch to separate them and treat them differently. Thus, the alternative approach of transmitting the different media separately, leads to a more general and flexible solution, albeit more complex. This solution introduces independence among the media, which has the potential to diminish the effects of communication errors.

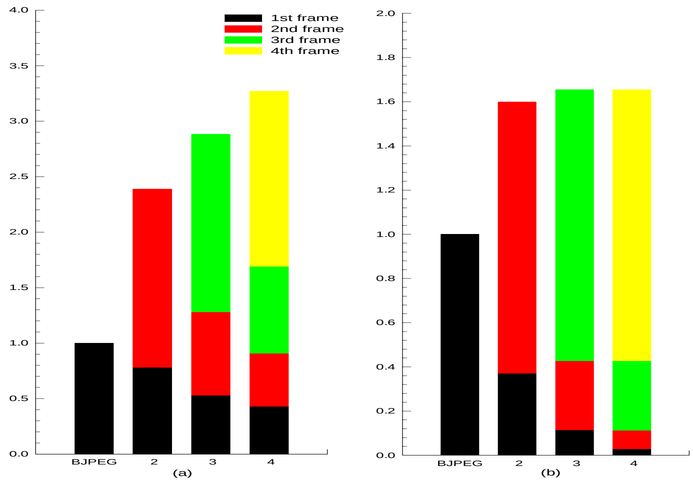


Fig. 4. Ratio of time to (a) compress and (b) decompress an n-layer HJPEG image (bridge).

Given a solution to the inter-media synchronization problem that allows for independent media streams, the extension to multiple layers, with each layer using a separate stream is straightforward. A survey of synchronization techniques is presented in [17]. The only concern then is the added complexity of synchronizing a larger number of streams.

A final observation is that in the case of interactive communication, the tight delay constraints can have the side effect of simplifying the synchronization problem by minimizing the time difference between any data that would be considered for presentation (i.e., delayed data that would be significantly out of sync would be dropped because they would be considered unacceptable).

B.3 Signal Processing Issues

It is well known that LC requires considerably more complex signal processing than non LC, quality and other parameters being equal, and thus it requires more computational power in order to carry it out in real-time. Our compression/decompression time measurements in section II-B.1 reiterate and confirm this. We believe, however, that as the advantages of MLC are recognized, the added cost of MLC vs. non-layered compression will be marginal.

Note that HJPEG, but probably MLC in general, has the desired property of asymmetric complexity: decoding is much faster than coding (e.g., see Fig. 4). Therefore, receivers, particularly on wireless mobile terminals can easily use and benefit from MLC. A related issue concerns compatibility across platforms, equipment type, and generation. It is not clear to what extent MLC can be successful, particularly in multicast applications, if most receivers are not compatible with it. In many such situations solutions of the type of "the least common denominator" prevail; we hope this is not the case with LC.

C. *Arguments for Using Layered Coding in Packet-Switching Networks*

Layered coding of images and continuous media can help diminish terminal compatibility problems, exploit differences in presentation formats and parameters, and can address network path heterogeneity, congestion control, transmission errors, as well as, improve efficiency for real-time communications. In this section we discuss how these advantages can be achieved. Specific solutions for various network architectures and their merits are further expanded in section III.

C.1 Congestion Control

Layered coding can increase the efficiency of high-speed networks. The main issue for high-speed packet-switching network architectures (e.g., ATM) is the significant congestion control problems that can arise due to the statistical multiplexing of very high burstiness signals [18]. A key property of any solution (that is not based on resource reservation at the peak rate of the sources) is the ability to shed load quickly without causing an avalanche of retransmissions of dropped traffic [19]. With LC of CM, when network congestion arises it is possible to drop the less important signal components without causing service interruption, and without the need for retransmissions, but instead with a temporary reduction in service quality (which may be unnoticeable in many cases). Since it is expected that real-time continuous media will constitute a significant proportion, if not the bulk, of the traffic, the capability of quickly dropping a proportion of that traffic can be a key to the solution to the overall congestion control problem.

An interesting observation is that the basic (e.g., low resolution) components, which are important for signal continuity, are highly compressible, constituting a small fraction of the total data rate required for signal transmission. A reasonable strategy then is to transmit the low resolution, high priority components through connections with explicit performance guarantees, and transmit the other components as "best-effort" traffic, without explicit guarantees (or require less stringent guarantees, for example in loss performance). Schemes proposed for providing explicit Quality of Service (QoS) guarantees are rather inefficient [20] or depend on the existence of low-priority, "best-effort" traffic in order to improve efficiency. Therefore, efficiency can be considerably increased if the portion of the traffic which requires stringent QoS guarantees is minimized and part of the signal can be transported as "best-effort" traffic.

One of the arguments against using LC to facilitate congestion control is that the overhead compromises the basic premise, by pushing congestion (and thus loss) to higher levels or by removing a bandwidth margin which could be used for Forward Error Correction (FEC). However, this is not a valid argument when both guaranteed QoS and "best-effort" channels are used because the main concern then is the number of connections that can be admitted with guaranteed QoS. This number is considerably increased when LC is used because LC allows applications that require high QoS to reserve resources, not at the level of their peak demands, but at a much lower level, by sending their additional traffic either as "best effort" or with much less stringent QoS requirements.

C.2 Multicasting Continuous Media

Traditional communication modes have been one-to-one, or unicast, and one-to-all, or broadcast. The generalization of these two extremes is multicasting, the communication of a single sender with a select group of receivers which may or may not include the sender. In multicasting, a sender communicates explicitly with a "group," rather than with individual actors which happen to be members of the group. Consequently, the sender need not be knowledgeable of group membership, and in fact, the group membership may change over time [21].

A number of dynamic control mechanisms in protocols employ feedback from the receiver to the sender. However, with multicast, how to provide feedback efficiently (or whether it should be provided at all) is a non-trivial question. There are four basic approaches:

1. Ignore the issue and, if problems arise, delegate the function to higher layer protocols.
2. Emulate multiple unicasts from the source, at least as far as control is concerned, with the source maintaining state information about each of the destinations.
3. Distribute the control mechanism over the nodes of a multicast tree (typically the same tree used for traffic distribution).
4. Take preventive action to minimize reliance on feedback. For example, for error control this means using FEC [22], [23] to anticipate errors and provide redundancy, allowing the receivers to reconstruct the information without requiring the sender to retransmit. For flow/congestion control, this means reserving resources so that receivers and intermediate switching nodes are always able to support the flow rate dictated by the sender or have mechanisms that allow control of congestion (by dropping traffic) which do not require the sources to retransmit.

The fourth approach seems to be the most appropriate for multicasting CM. The desirability of this solution arises from the lack of complexity due to feedback control mechanisms, even though it is an improvement over simply doing nothing. The cost is the additional complexity due to these anticipatory schemes.

A communication abstraction promoting open-loop control and providing a service analogous to that of a television broadcast channel is the *Multimedia Multicast Channel* [24]. A source transmits CM streams onto the channel and receivers "tune in" to the channel to receive a selected subset of the streams. To support heterogeneity, each receiver may tailor the selected streams to meet individual needs through the use of filters, which, if compatible, can propagate upstream and combine in order to economize on resources [25].

C.3 Terminal Heterogeneity

Many types of (multimedia) terminals and media representation standards exist and it is expected that the heterogeneity will persist. This has also been the case for traditional text-based computer communications, where the compatibility problems are considerably simpler than for multimedia. The presentation layer in the OSI architecture addresses the issues of data representation and compatibility problems. When incompatible transmitters and receivers communicate, translation is necessary. This service can be provided in one of three places: (1) at the transmitter, (2) within the communications network, or (3) at the receiver.

If the third solution is possible, there is no real compatibility problem. The second approach is the typical solution (e.g., traditional protocol converters). The first solution is similar to the third in the point-to-point case, but is rather inefficient in the multipoint case because it requires the sender to translate its data format once for every incompatible receiver. This translation consumes sender resources that are usually highly contended and requires the network to transport a higher volume of information because sharing of links is not possible.

Facilitating translation at the receiver through synthesis of the signal from separate components can be achieved through LC. In addition, admission control and routing algorithms may take into account the different needs and capabilities of the destinations by, for example, forwarding only usable components to select destinations [6]. Furthermore, there is an interesting interaction between LC and multicasting. MLC enables destinations to adjust the quality of the signal they receive each, independently and without the source actually being aware of this adjustment (of course, the adjustment is only possible towards lower quality). This is a very important property considering the feedback control problems of multipoint communication.

C.4 Error Control for Wireless Networks

Errors in transmission can be handled at different levels. In point-to-point computer network links, error-free communication is usually achieved by error detection and retransmission. Error control is particularly difficult with multipoint connections because of the feedback related problems (e.g., ACK implosion, conflicting requests, etc. [21]).

Physical layer error control corrects errors using block or convolutional coding. Essentially, channel error control techniques can be thought of as reducing noise levels. These schemes are media transparent and errors are corrected perfectly as long as the error rate is within the design range of the codes applied, but they seem to perform poorly for video transmission over wireless links [26]. Some of their drawbacks are the following. First, it is impractical to correct all errors, therefore they cannot be considered by themselves complete solutions, particularly for wireless networks which are highly noisy. Second, FEC introduces considerable overhead in bit rate. Third, protection (and the associated overhead) is applied uniformly to all traffic over the link, treating media with different QoS requirements alike.

The second method is to use retransmissions based on feedback from a receiver in order to correct errors. An error detection code is used and, if an error is detected, the sender is asked to retransmit until the receiver acknowledges a correct reception.[5] However, Automatic Repeat reQuest (ARQ) protocols are generally inappropriate for interactive multimedia communications due to the delay they introduce, at least end-to-end for "global" communications. In addition, because the number of retransmissions is unpredictable, ARQ introduces jitter and unbounded/unpredictable delay, as well as overhead.

An important characteristic of real-time interactive applications such as videoconferencing is that the applications can tolerate some errors without significant degradation in quality. The consequence is that even without fully reliable transmissions, applications can be made to recover gracefully from loss of segments (at least for some media components). There is, thus, a tradeoff for real-time continuous media between smooth progress, ensured by avoiding reliance on feedback, and complete reliability, for which feedback is necessary. Various technologies and solutions choose different operating points in this space. For example, most of the Internet/Mbone based conferencing applications avoid the use of the reliable, connection oriented TCP, relying instead on the connectionless UDP.

The error tolerance of many image based applications, the difficulties with feedback, and the interest in application level robustness, are the reasons behind a third method of handling errors in picture transmission by concealing the damaged portion of the image after reception. This method is suboptimal in the sense that the damaged region can be at best remedied rather than corrected. However, for real-time multimedia transport, delay considerations are usually more important than loss of some (enhancement) detail, due to human perception characteristics. Advantages of this approach over the schemes based on fully reliable data transfer are that (1) potentially, no additional bandwidth is necessary for control codes (although some ef-

[5] NAK based schemes can also be used and are more appropriate for multipoint communications.

fective concealment methods introduce overhead, e.g., repeated or "protected" transmission of motion vectors for video, and (2) there is no specific upper bound on the number of errors, beyond which catastrophic failure results, even though a large number of errors can render the concealment schemes ineffective. Therefore, concealment schemes can be used as a last resort to improve error resilience.

Another approach to cope with network errors is LC. Albeit their inherent error-resiliency, LC solutions have a well-known drawback in the form they are typically proposed: they require a costly or often infeasible guaranteed performance channel for the base layer carrying the critical subsignal, in order to maintain some minimum picture quality. Furthermore, as we will see in section 4, LC may suffer more than non-layered coding from packet loss when the base layer(s) cannot be protected.

Note that the problems that LC must protect against in high-speed fiber-based ATM networks are very different from those encountered in unpredictable, highly dynamic wireless networks and links. The former have extremely low bit error rates and therefore cell and packet loss will be mostly due to buffer overflow leading to error bursts due to congestion at network switches. Note however that, as discussed previously in section II-C.1, this cell loss with the proper labeling can be controlled in order to be limited to enhancement layer traffic. However, this cannot be the case when the loss is mostly due to errors during transmission over the air, which cannot be controlled at the level of individual cells or packets.

Thus, there are two potential ways in which LC, and MLC in particular, can be beneficial for network error control. First, as we will see in section 4, even in the absence of special protection mechanisms for the most important image layers, the degree of error resilience can be moderate with MLC. This is mainly due to signal decomposition and scrambling. Second, because errors have different impact at the various layers (as we will see, again, in section 4), the level of protection provided to each layer, through FEC (e.g., the PET scheme [23]) or other means can (and should) be tailored to the importance of the layer in the final image quality. The result of this added protection is that the *residual* error-rate is different for the various layers. The benefits of the approach are mainly due to the fact that lower layers are more important for the final image quality, but usually significantly smaller in size (or rate), therefore, strong protection can be cost-effective.

III. Transport Architecture for MLC Media

The advantages of LC are best exploited when the networking architectures support differential treatment of traffic. Even though priorities and labeling for different types of traffic have been a part of the Internetworking Protocol (IP) since its original design [27], they have seldom been implemented in network system software or used in production mode on the Internet. The situation has been similar for most other wide-area networks. Local Area Network (LAN) specifications, notably FDDI, have been at the forefront of supporting priorities and Quality-of-Service (QoS) guarantees. However, in this case the impact of QoS guarantees for real-time imaging is probably less critical due to the relatively high bandwidth available on LANs and the small propagation delay.

Productive use of priorities and service differentiation still seem to be in the future of networking, with both ATM and the next generation of IP (IPv6) expected to use traffic differentiation in order to provide improved QoS and/or to guarantee it. A general framework for the dissemination of real-time CM through packet switching networks is the Multimedia Multicast Channel (MMC) [24], [25]. The MMC design is independent of specific technologies, such as IP or ATM, but compatible with MLC and network architectures that provide services with different QoS guarantees.

A. ATM

One of the main reasons why considering more than two layers for LC has been avoided in the past, in addition to the various reasons we discussed in section 2, is a lack of explicit support for MLC in ATM and the existence of an explicit mechanism supporting two-layer coding, namely the use of the Cell Loss Priority (CLP) bit in ATM cells. However, with the increasing interest in wireless ATM and heterogeneous networks, it is appropriate to consider more general mappings between the requirements and capabilities of various technologies.

The traditional mapping of LC to ATM service has been through the use of the Cell Loss Priority (CLP) bit in the ATM cell header, as defined in the ATM specification (e.g., see [28]). ATM switches that experience congestion are expected to first drop cells that have the CLP bit set. This action should relieve congestion and thus protect cells that have not been designated as candidates for dropping. Obviously, this technique is only effective if a significant amount of traffic with the CLP bit set can be found at a switch at times of congestion.

Two mechanisms have been suggested for setting the CLP bit. First, the network at its entry points can set the CLP bit when incoming traffic violates its traffic contract with the network, i.e., as a soft implementation of a policing function. This mode of operation is content independent and either irrelevant or possibly detrimental to LC of images and continuous media. The second mechanism depends on a source designating some cells as less important and candidates for dropping, based on their information content. This is exactly where a match between LC and the ATM specification exists, because designating some cells as useful but non-critical is fully compatible with the definition of LC.

The CLP bit approach has some drawbacks and is not flexible enough given the various applications of LC that one can envision. First, there is direct support (differentiation) for only two layers. Second, without further service specification the QoS provided to the two layers, and particularly to the layer using cells with the CLP bit set, is not known. Therefore, we propose the following alternative

Fig. 5. Transport of a MLC signal over multiple virtual channels in a heterogeneous ATM network. S is the source, A, L, H, and W are receivers (adaptive, low quality, high quality and wireless), S1 – S5 (ATM) switches, N1 – N5 (ATM) networks with N5 a wireless network, and B a base station. The number of concentric circles for the source and receivers represent terminal capabilities or user quality settings.

This proposal has some specific implications for ATM switch designs, which we discuss next, but we believe it is in the mainstream of the rationale for the adoption of ATM and the implementation of B-ISDN. First, it relies on switches using per-VC queueing and forwarding. However, this seems to be the trend in new ATM switch offerings. Second, it increases, by a factor equal to the number of layers, the number of VCs that switches and network controllers need to service and manage. This is a potential drawback of the approach, but we believe it is minor compared with the advantages it offers, particularly in the multipoint case where separate connections might have to be maintained for receivers with incompatible settings. Furthermore, ATM provides a mechanism to address this problem in core switches where it is more critical because of the traffic aggregation: the Virtual Path (VP) mechanism. A VP provides a way for ATM switches that are not the final destination of traffic to treat and switch traffic from multiple VCs collectively. For example, in Fig. 5, S2 "sees" a VP going through it where the VCs in question are contained, but is unaware of their detailed set-up. The only constraint is that the QoS offered to the VP must be able to satisfy the QoS of the most stringent VC it carries. Alternative designs, e.g., having more than one VP with different QoS levels, in order to decrease the "cost" of less demanding traffic at core switches, are easy to arrive at.

A major advantage of this approach is that a receiver does not need to *receive* all the data transmitted by the source, but instead it can select to "subscribe" (connect) to only the VC that it is interested in. This is extremely significant for bandwidth limited wireless mobile terminals uninterested in, for example, high-resolution components of the signals, participating in multicast sessions, as shown in Fig. 5. In that case, the high-resolution signal does not need to be transmitted over the air and the source needs not be aware of it. Furthermore, the base station, B, can be simple and completely unaware of the media content it carries (instead of a protocol/media converter that would otherwise be required). Finally, depending on the available signaling protocols, adjustments to signal quality could be made easily and dynamically by adding or dropping connections. Note that the demultiplexing cost for receiving multiple VCs arriving at a receiver (as opposed to the signal processing cost for synthesizing the final presentation signal from its components) is minimal based on our experimental findings reported in [29].

B. Internet

Similar techniques can be applied in IP networks. The current IP specification (IPv4) has two (related) mechanisms that can be used for traffic labeling and prioritization: the type-of-service field and an explicit priority field that allows eight levels of priority to be specified.

In addition, even if the sources themselves do not explicitly specify the type of traffic, some service differentiation could be achieved for applications that use the UDP or TCP transport layer protocols by having routers snoop at the UDP/TCP port numbers. This assumes no fragmentation, but fragmentation is now considered a bad idea anyway. A method for specifying the service equivalent of a connection in a connectionless network such as the Internet, without support or involvement of the applications, is the use of (implicit) flows, as described in [30].

Explicit flows, on the other hand, are included in the specification for the next generation of IP (IPv6), through a new IP packet format [27], [31]. In addition, a new priority structure, potential support for reservations and QoS guarantees, etc., are introduced with IPv6 and related technologies such as RSVP [32] and others.

IV. Error Control and Concealment

A. Past Work on Error Concealment

Many compression standards such as JPEG [9], MPEG [33], and H.261 [34] and its variants, use the Discrete Cosine Transform (DCT) applied to blocks of image samples. The coefficients after the DCT are then quantized and entropy coded in order to increase the compression efficiency. Packet loss during transmission can damage a sequence of blocks and this is exacerbated when the coding scheme uses run-length encoding, as most schemes do.

Numerous techniques for error concealment have been developed, all of them consistently exploiting the spatial and possibly temporal redundancy of the picture components [3]. In intracoded frames, spatial interpolation methods are used to conceal errors, replacing pixels in damaged blocks with interpolated pixel values from surrounding blocks. This method, however, assumes that neighboring blocks are not affected from the packet loss and it alone cannot effectively conceal clustered block loss resulting from long error bursts.

To alleviate this problem, block deinterleaving and scrambling were proposed and studied [35], [36]. Separation of the data stream into multiple streams (deinterleaving) achieves the dispersal of the DCT coefficients throughout the image, improving the error resilience of the stream. However, either the resulting image quality is inferior to the one without deinterleaving when no errors occur, or a reduced coding efficiency must be accepted in order to achieve the same quality. The scrambling method separates spatially adjacent blocks so that the impact of packet loss is reduced. However, an increased bit rate follows due to the lower coding gain resulting from the less precise prediction values of this approach. A 8%-15% overhead is reported in [36].

In interframe coded video, the use of motion vectors for concealing errors is common [37], [38]. This approach can be very effective when the estimation of motion in subsequent frames can be achieved accurately. Thus, one of the standard assumptions used in the literature is that motion vectors will be delivered error-free. This of course is a rather strong assumption and not always easy to guarantee.

Despite considerable efforts to perfect concealment schemes exploiting spatial and temporal redundancies for non-layered coding, only a few efforts have been reported specifically for LC, most concentrating on video. Often the need for concealment is ignored [5], [7] due to the inherent error resilience of LC with a protected base layer. Indeed, most authors assume that the base layer can be adequately protected. However, errors in the base layer cannot be effectively concealed in the absence of a reference picture and accurate motion vectors. A simple zero or mean substitution will not be effective for a damaged base layer [36]. Further, in the case of hierarchical coding with variable spatial resolutions, these techniques usually aggravate the situation.

B. Decoder Resynchronization

With variable length coding, loss of even a single packet can lead to loss of decoder synchronization, which can damage an image catastrophically. Synchronization flags, usually preceded by a definition header, can be inserted around entropy coded segments in order to identify them as such without the need to decode the compressed data. The definition header contains the specification of a restart interval, which is an integer multiple of the minimum coded unit, e.g., the block. Various error conditions such as missing markers or out of range values trigger error recovery procedures at the decoder. When the decoder detects an error condition, it scans for the next flag in order to resynchronize by resetting the decoder. The relative frequency of restart flags can be increased leading to increased robustness at the expense of coding efficiency; this is a basic design trade-off. Note, however, that recovery is not possible from all error conditions; in particular, header information loss is catastropic [9].

The impact of the loss of synchronization with various coding schemes is depicted in Fig. 6. A restart flag every 8 lines is used in this case. We observe that with proper use of resynchronization schemes, the decoding process can maintain synchronization, despite errors leading to block loss. Note also that use of LC is advantageous.

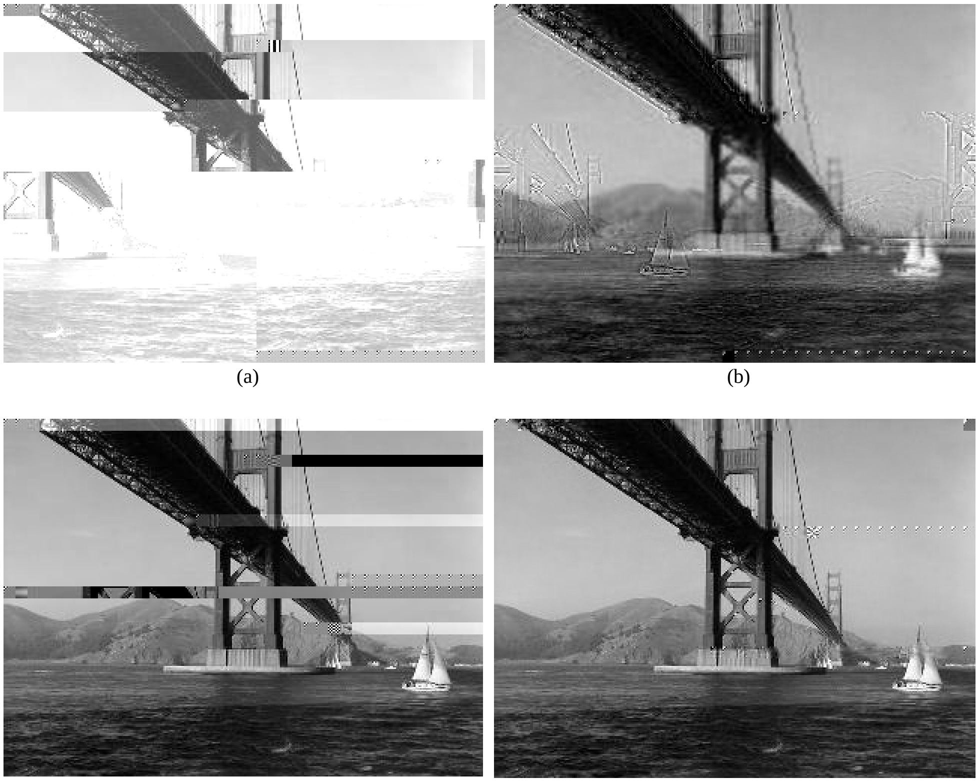
Alternatively, the header field of a packet can convey direct addressing information, thereby reducing the chance of catastrophic synchronization error, particularly since the number of pixel blocks within a packet is rather small. Furthermore, in the case of interframe coded video using conditional replenishment, as packets update blocks at specific locations excluding stationary regions, a short restart interval is naturally attained. Hence, in these cases the impact of synchronization errors is usually confined.

C. Error Resilience of Multi-Resolution Layered Coding

Here we evaluate the error resilience of MLC considering the possibility of errors at the base layer or, more generally, the lower layers of a MLC image and potentially multiple levels of protection, adjusted to the importance of each layer. Depending on the situation, the different protection levels can be achieved through various techniques. For example, multiple QoS levels or priorities could be used if the problem is switch congestion leading to cell or packet loss, or different degrees of FEC can be applied if the loss is due to transmission errors.

First, note that by the definition of MLC, even complete loss of the last layer is secondary or can even be insignificant or unnoticeable. This is illustrated in Fig. 2. The original image, shown in Fig. 2 (d), is coded and transmitted as a 4-layer MLC image. Any errors or packet loss occurring in layer 4 data can be masked by displaying the decoded image including only layers 1–3. The result is a lower resolution, but otherwise perfect image, as shown in Fig. 2 (c).

This example demonstrates the basic error resiliency of LC. Note that up to roughly 2/3 of the size of the 4-layer



(a) (b) (c) (d)

Fig. 6. 2.5% loss in (a) BJPEG without restart interval (b) BJPEG with restart marker every 8 lines, (c) 4-layer HJPEG without restart interval, (d) 4-layer HJPEG with restart marker every 8 lines. The HJPEG images are fully protected up to the third layer.

coded image can be lost with little impact on image acceptability. Furthermore, if errors occurred even in layer 3, that layer can be dropped too, resulting in the image shown in Fig. 2 (b). There is now noticeable degradation in quality from the loss of the low spatial frequency signal, but still the image is probably preferable to many images with much less data in error but no LC or sophisticated concealment as shown in Fig. 6.

Thus, terminating the decoding early, i.e., disregarding all received information from the last layer (or last few layers), is the simplest, but also a very effective error control strategy when errors occur in the higher layers. Of course, refined versions of this strategy can be used when the errors are concentrated on part(s) of the image, leading to an image that is identical to the original, except for a reduced resolution in parts of the picture.

Block loss in intermediate layers can be tolerable depending on their levels in the image hierarchy and the loss patterns. In general, the impact of errors and loss in the lower layers is far greater than that at higher layers (see Fig. 7) and packet loss crossing multiple layers is more detrimental. Given these observations and a fixed budget for error correction overhead, it becomes apparent that multiple protection levels, with higher protection for the smaller but more significant lower layers, would be beneficial.

Fig. 7. Impact of errors at the lower layers. From left to right: original *Lena* image, and 6-layer MLC image with all blocks lost up to the first (0.07% of total size), second (0.37%), and the third (1.54%) layer. Each MLC image is obtained by aggregating differential layers with neutral gray background.

C.1 Error Performance of MLC with Unequal Protection Levels

For our experiments described here we use the luminance components (only) of the following two image sequences: *Football*[39], in almost CIF resolution (360×243 for luminance), and *Salesman*[40], in CIF standard resolution (352×288 for luminance). Each test sequence has 80 frames and is coded with the HJPEG coder we have implemented [10].

In these experiments, packet loss is assumed to be random throughout the affected layers in each picture frame.

TABLE I

Distribution of block loss among layers

Experiment	Coding method	Residual loss rate (% of raw)			
		1st layer	2nd layer	3rd layer	4th layer
Case I:	2-layer	50	0	—	—
Zero substitution	3-layer	25	50	0	—
(crude concealment)	4-layer	12.5	25	50	0
Case II:	2-layer	**50**	0	—	—
Temporal replacement	3-layer	**25**	50	0	—
(moderately effective)	4-layer	**12.5**	25	50	0
	2-layer	**33**	0	—	—
	3-layer	**11**	33	0	—
	4-layer	**5.5**	11	33	0
Case III:	3-layer	0	50	0	—
Protected base layer,	4-layer	0	25	50	0
zero substitution for others					
Case IV:	2-layer	**50**	100	—	—
Temporal replacement	3-layer	**25**	50	100	—
(very effective)	4-layer	**12.5**	25	50	100
	2-layer	**33**	100	—	—
	3-layer	**11**	33	100	—
	4-layer	**5.5**	11	33	100

However, in order to make a fair comparison among the various test sets we use the same seed to initialize the pseudo-random sequence. In order to focus on concealment techniques and performance, header information and tables are assumed to be protected in order to avoid complete failure of displaying the image[6] and loss of decoder synchronization is not allowed. This enables us to quantify residual prediction error from the lower layer blocks without being distracted by the artifacts introduced by the loss of synchronization. Table I summarizes the set up of the experiments (boldface represents the blocks that are candidates for temporal replacement in the presence of block loss). The loss is assumed to occur in the first $n-1$ layers of n-layer coded images in Cases I,II and III (i.e., we assume that there is no loss in the top layer) and in all n layers in Case IV. The peak signal-to-noise ratio (PSNR) is used to measure the image fidelity objectively. The PSNR is defined as $10 \log \delta^2$ / MSE where δ is the peak pixel value (255 for 8 bits/pixel image) and MSE is the mean square error between the original and the reconstructed image.

To simulate a crude error concealment scheme for errors or loss at the base layer (Case I), we substitute zeros for lost pixel blocks. Fig. 9 shows 3–6 dB difference in PSNR between 2-layer and 4-layer coding. In this case, the base layer is protected so that the *residual* error rate is 50%, 25% and 12.5% of the raw error rate for 2-layer, 3-layer and 4-layer coding respectively. This result demonstrates that when loss occurs at the base layer, MLC with more layers (3-layer and 4-layer in this case) with different degrees of protection across layers is more effective in mitigating the impact of errors than ordinary 2-layer coding, if a concealment mechanism is not used or is poor.

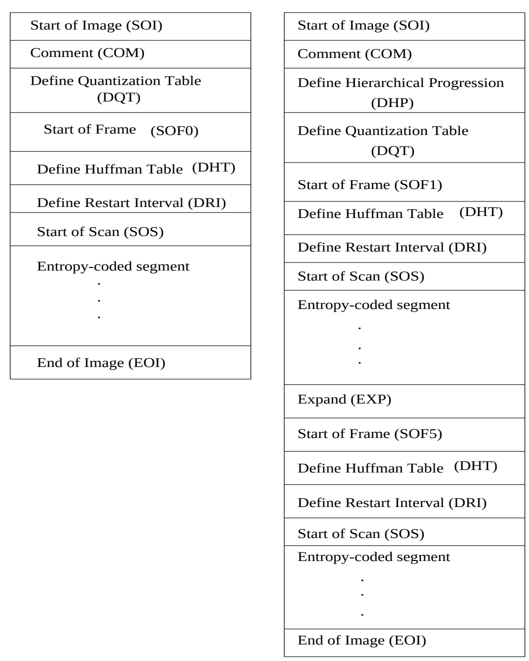


Fig. 8. Compressed data structure for DCT-based (a) BJPEG (non-layered), and (b) 2-layer HJPEG (sequential Huffman).

The *football* video sequence has moderate and intermittent fast motion and is used to simulate a moderate error concealment scheme for errors or loss at the base layer (Case II). Temporal replacement, i.e., replacing the lost block with the corresponding block in the previous frame, is

[6] The range of missing packets was confined to image data after the *start of scan* (SOS) header, i.e., only entropy coded data is assumed to become targets for packet loss–see Fig. 8.

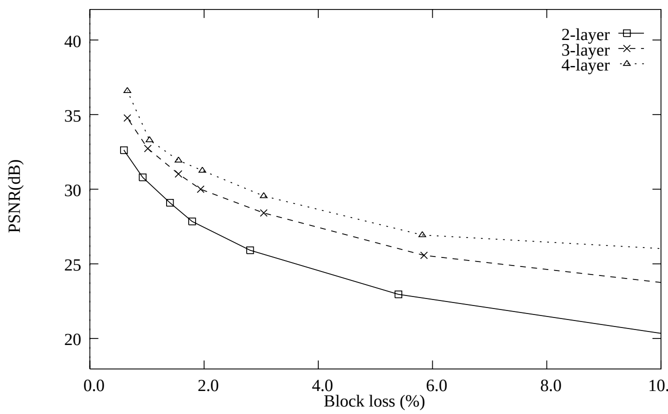


Fig. 9. Effect of block loss in n-layer images with absent or crude error concealment. The residual error rate at the base layer is 50%, 25% and 12.5% for 2-layer, 3-layer and 4-layer HJPEG, respectively. The results are obtained using the *Football* sequence.

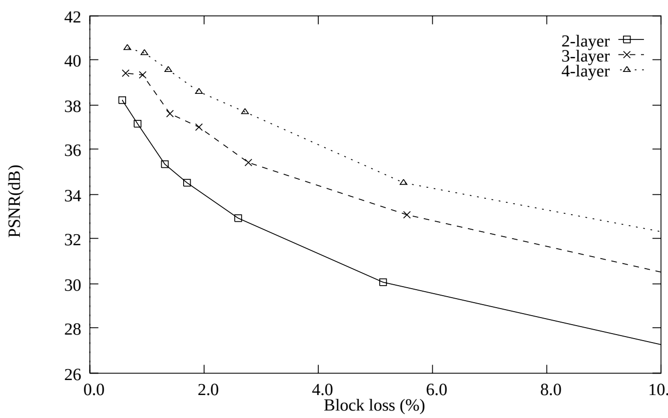


Fig. 10. Effect of block loss in n-layer images with moderate concealment. The residual error rate of base layer is 50%, 25% and 12.5% for 2-layer, 3-layer and 4-layer HJPEG, respectively. The results are obtained using the *Football* sequence.

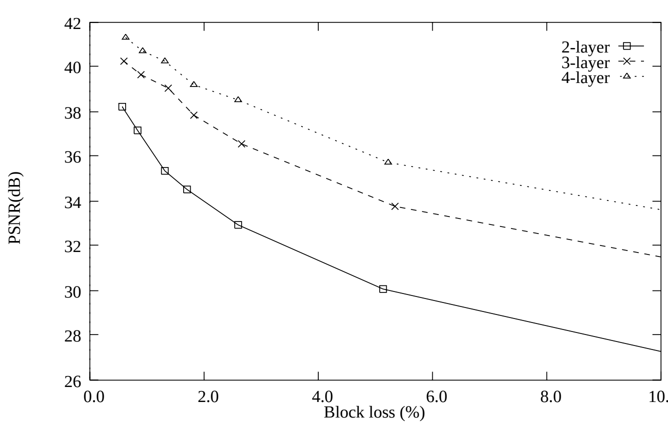


Fig. 11. Effect of block loss in n-layer images with moderate concealment. A higher protection is provided to the lower layers (residual error rate of the base layer is 33%, 11% and 5.5% for 2-layer, 3-layer and 4-layer HJPEG, respectively). The results are obtained using the *Football* sequence.

TABLE II

AVERAGE INTERFRAME CORRELATION INDEX ACROSS MULTIPLE LAYERS IN THE *Football* SEQUENCE.

Encoded sequence	PSNR (dB) 1st layer	2nd layer	3rd layer	4th layer
2-layer	24.03	23.49	—	—
3-layer	24.93	24.02	23.49	—
4-layer	26.30	24.92	24.03	23.49

used to conceal errors. Every other picture frame in the sequence is subject to loss (in order to make the replacement scheme effective), but the loss pattern is random within the frame. Since the sequence is only intraframe coded, no motion compensation is applied and hence the effect of error propagation in subsequent picture frames is not considered.

Fig. 10 shows the PSNR as a function of the overall error rate in n-layer coded images. The figure shows a PSNR gain of 3–5 dB over 2-layer coded sequence, which also agrees with a better image sequence quality in subjective evaluation. Temporal replacement is used only for the base layer. This concealment method performs rather poorly when there is high speed motion in the scene, which is the case for the early part of the sequence. The impact is evidently more severe with 2-layer coding. We have observed that using temporal replacement for lost blocks in higher layers of MLC image sequences results in almost the same or even slightly lower PSNR than 2-layer coding. This can be explained in part by the lower interframe correlation of the higher layers. Table II shows average interframe correlation indices for the *football* sequence. We use PSNR values between two adjacent frames as an Interframe Correlation Index (ICI), defined as

$$ICI = 10\ log\ \delta^2\ /\ MSE_{f-1,f}$$

and

$$MSE_{f-1,f} = \frac{1}{M \times N} \sum_{m}^{M} \sum_{n}^{N} |E_{f-1}(m,n) - E_f(m,n)|^2,$$

where δ is the peak pixel value and $|E_{f-1}(m,n) - E_f(m,n)|$ is the pixel differential between the two adjacent $M \times N$ frames, $f-1$ and f, of the video sequence.

Since pixels are significantly uncorrelated in differentially coded layers, direct concealment either spatially or temporally will not be effective for damaged blocks in enhancement layers. Note that the superposition of the subsequent enhancement layer to the base layer in an n-layer coded image is equivalent to the base layer of the $(n-1)^{-st}$ layer image in its quality. As the degree of error protection increases, MLC with more layers yielded higher PSNR, as expected (see Fig. 11).

In Case III, full protection is granted to the base layer and thus we assume that no errors occur at this layer. Instead, we view the second layer of 3-layer coding, and the second and third layers of 4-layer coding, as representing "base" layers (i.e., they are basic, non enhancement layers for the image), and we assume that they are subject to loss. This setting does not exactly correspond to the image hierarchy since most of the image energy is already in the real base layer and the one or two "base" layers are differential layers with error predictions based on the first layer. Even so, this setting is useful in that at least we can examine to extent the interplay of different degrees of protection across layers.

Fig. 12 demonstrates that the PSNR performance of MLC with different degrees of protection is actually opposite to Case I when the error concealment is presumably very effective. The lower PSNR from 4-layer MLC is mainly due to the overall low ratio of protected blocks in the simulated base layer (i.e., fist enhancement layer) as compared to 3-layer coding rather than coding characterics. The base layer is already enough of an approximation of the image (i.e., the simulated concealment performs extremely well) and no loss patterns or error rate changes in subsequent layers are likely to affect the image quality significantly. Hence the effect from different coding and protection strategies is minimal. However, due to the nature of the simulation, no distinction among different coding methods could be observed in subjective evaluation.

Another simulation of a very effective concealment scheme is performed using a part of the *Salesman* sequence with slow motion (Case IV). In this case, the base layer is protected so that the *residual* error rate is 50%, 25% and 12.5% of the raw error rate for 2-layer, 3-layer and 4-layer coding respectively. The *residual* error in the last enhancement layer of each n-layer coded images is 100% of the raw error rate. Temporal replacement is used only for the base layer (as in Case II). The concealment method is very effective in this case since far less motion is present in the scene, which results in a higher PSNR for 2-layer coding (see Fig. 13). This is in part due to the fact that affected blocks in the stationary background are copied from the corresponding blocks in the previous frame perfectly in 2-layer coding, while interpolated blocks are used for damaged blocks in MLC with more layers.

This suggests that concealment through interpolation from lower layers is effective when only spatial redundant information is available as in still image transmission. For video transmission, MLC with more layers does not improve the existing error resiliency of layered coding because the temporal replacement scheme with protected motion vectors is usually very effective. Under high error rate conditions, MLC with more layers can be more fragile than 2-layer coding due to error propagation between layers within an image. This accumulation of error from lower layers cannot be ignored, especially when artifacts are introduced from the loss of synchronization.

From Fig. 14, we can observe that the PSNR gap between different coding methods is reduced albeit slightly, as higher protection is provided to the lower layers (33%,11% and 5.5% *residual* error rate of base layer for 2,3 and 4-layer coding, respectively). In visual evaluation, however, substantial improvement on image fidelity was observed with MLC with more layers. Subjectively, loss in multiple layers (i.e., overlapping error of a layer and the propagation error from its previous layer) other than the last two layer in spatial hierarchy is visually unacceptable. Although actual picture quality largely depends on the overall residual error rate which dictates the degree of required protection level for lower layers, the degree of different protection level is not exactly linear to the increase of error present in image. This suggests that a same degree of high protection should be applied in lower layers of image where channel error rate is higher than some statistically determined thresholds. The protection boundary does not necessarily coincide with the layer boundary in these layers.

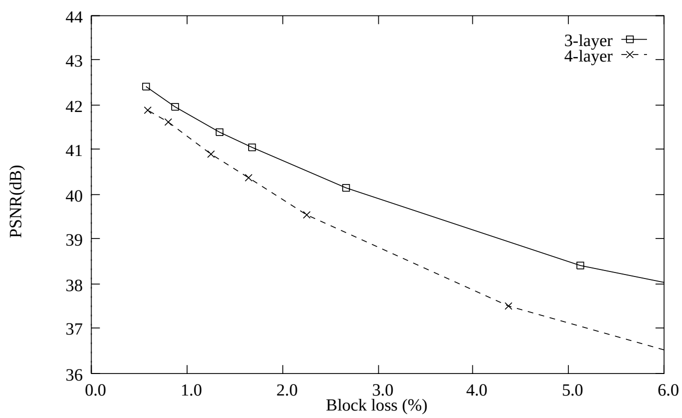


Fig. 12. Effect of block loss in n-layer images with the base layer protected and the second layer is treated as a base layer. The residual error rate for the second layer is 50% and 25% for 3-layer and 4-layer HJPEG, respectively. The results are obtained using the *Football* sequence.

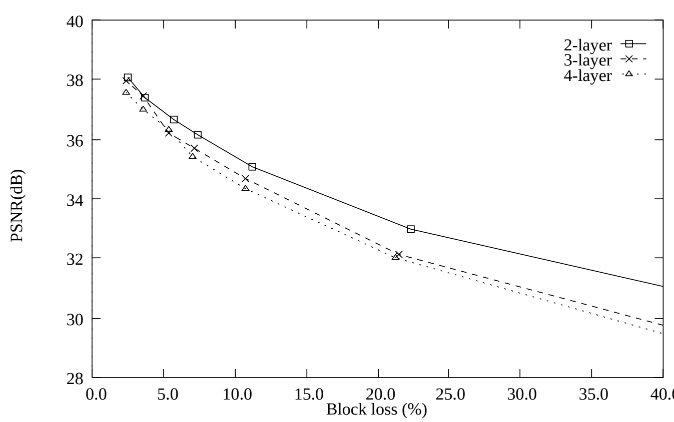


Fig. 13. Effect of block loss in n-layer images with almost perfect error concealment. The loss rate includes block loss in the last enhancement layer. The residual error rate at the base layer is 50%, 25% and 12.5% for 2-layer, 3-layer and 4-layer HJPEG, respectively. The results are obtained using the *Salesman* sequence.

In the case of real-time image transmission, as the importance of the base layer and layers close to it intensifies as we have more layers, it is critical to protect these layers to the highest degree possible. Fortunately these layers constitute only a small fraction of the whole image. Note, however, that simply increasing the number of layers in

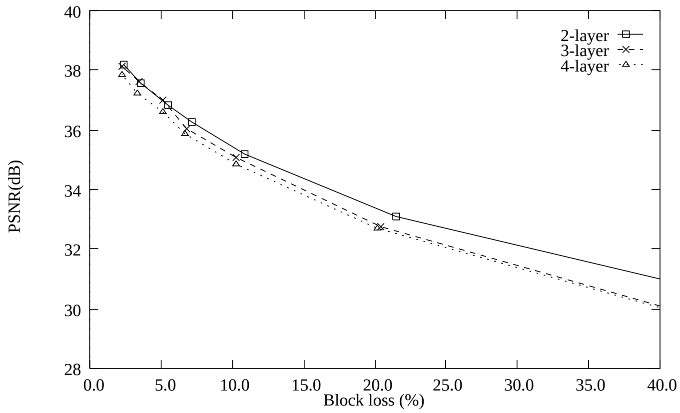


Fig. 14. Effect of block loss in n-layer images for almost perfect concealment. The loss rate includes block loss in the last enhancement layer. A higher protection is provided to the lower layers (the residual error rate at the base layer is 33%, 11% and 5.5%, for 2-layer, 3-layer and 4-layer HJPEG, respectively). The results are obtained using the *Salesman* sequence.

order to reduce the amount of data that needs protection (and thus the overhead) by protecting only the first layer, does not work well.

In the absence of a sophisticated error concealment scheme for the base layer, MLC with different protection levels can be effective when the lower layers are adequately secured. The error resiliency of MLC (with more than 2 layers) cannot be easily maintained otherwise. Due to the inherent recurrent structure of pyramidal coding, artifact-free image fidelity in coarse resolution layers is more important than in the higher layers. Therefore, various trade-offs exist between the number of layers used, the amount and type of protection applied to each layer, the resynchronization interval, the types and distribution of errors to guard against, the packetization method and parameters, and the delivered image quality. Further research is needed in order to better characterize this space.

As a last and extreme case, we consider full-layer MLC. This is MLC where the number of layers employed is such that the base layer is a single block (assuming an image of appropriate dimensions–see Appendix A). Although it is possible to layer up to a base block (8×8 pixel block in JPEG/MPEG), full layering is not necessary in practice.

Fig. 15 illustrates the effect of evenly spaced (non-random) errors leading to loss on two different coding techniques: non-layered JPEG and full-layer MLC. For non-layered JPEG concealment using mean pixel values from the closest pixels in undamaged neighboring blocks was applied.

In Fig. 15 (a) the block loss rate is 10.9%. Black blocks in the left image show the positions of block loss. The second image shows the reconstructed non-layered JPEG image after mean substitution. Artifacts are clearly visible in this case. The last image uses full-layer MLC with a simple zero substitution. It shows slight blurring, which is less annoying in subjective evaluation. Due to the impact of considerable block loss in the lower layers of MLC, full-layer MLC images often display dithering artifacts and blurring, which decrease the PSNR to a level just comparable to, or sometimes and particularly at low loss rates even lower than, non-layered images.[7]

With block loss stretched through multiple blocks at the increased rate of 25% shown in Fig. 15 (b), a better image quality is obtained with full-layer MLC. In this case full-layer MLC shows less artifacts than the non-layered image with mean substitution, albeit with more blurring. The objective measure, PSNR, did not indicate a much improved image fidelity. Observe that in the case of block loss in and/or near the bottom layer, the image loses much of its low spatial frequency signals and texture richness.

V. Conclusions

We have discussed how various features of Layered Coding (LC) can be applied in order to improve the responsiveness of image-based applications and the efficiency of packet switching networks transporting images and video in real-time. Congestion control, multicasting of continuous media (CM), terminal and network heterogeneity, and error control for wireless networks are key areas for the application of LC techniques in order to solve critical networking problems. For all but the first application, we argued that Multi-resolution Layered Coding (MLC) has significant advantages over the traditional two-layer coding schemes. The adoption of an open-loop control approach, suggested by the real-time character and the stringent delay constraints of interactive CM and made possible by MLC, also solves the multipoint feedback control problem.

We have also presented architectural alternatives in the context of various packet switching networking technologies, that can effectively support and exploit the features of LC, and MLC in particular. For transport over ATM networks, we proposed to adopt multiple ATM virtual channels with different QoS levels, instead of the standard two-level priority scheme based on the CLP bit. For the next generation of the Internetworking Protocol, IPv6, flow IDs can be used to efficiently achieve the same goal through IP routers. Furthermore, in order to illustrate the arising tradeoffs, we have presented experimental results for various coding metrics, such as compression ratio and coding and decoding times, obtained with the Hierarchical JPEG (HJPEG) codec (based on the existing JPEG standard) which we have implemented.

Finally, examining the performance of MLC in the presence of errors leading to packet loss possibly impacting all layers in the image hierarchy, we showed the importance of appropriate, in general different, degrees of protection for the lower layers. The error resiliency of MLC is particularly important when feedback-based error control is not feasible or cost-effective. Note that MLC can provide a reasonable error concealing effect through full-layering at low error

[7] On the other hand, this kind of error pattern, single isolated block loss, is unlikely in practice since any compression method using differential pulse code modulation (DPCM), where an erred DC coefficient of the impaired block renders all subsequent DC coefficients unusable until the next synchronization unit. In case of JPEG, restart markers (RST) are used to encode and decode each code segment independently of other intervals in the scan.

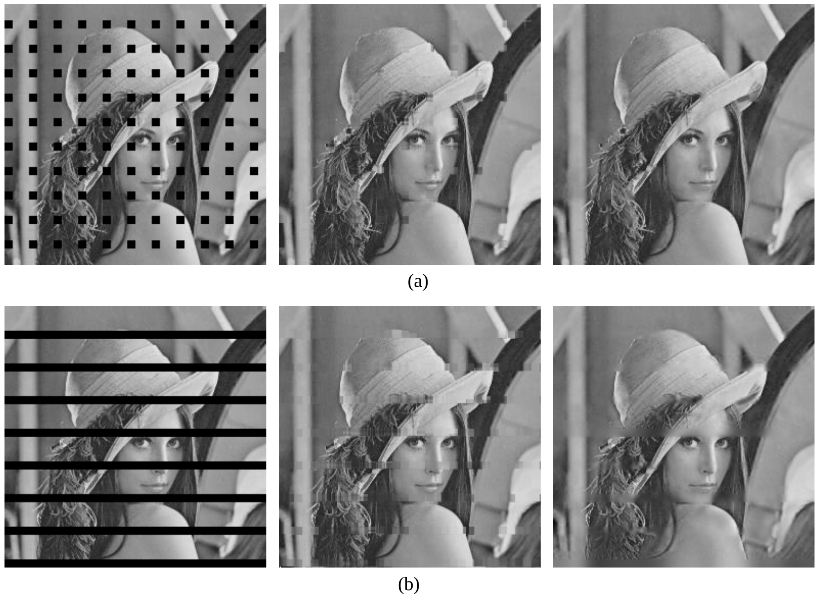


Fig. 15. Images with (a) 10.9% evenly distributed block loss and (b) 25.1% strip block loss. From left to right: Loss marked as black, non-layered coding with mean substitution, and MLC image with zero substitution. Note that there is no block loss in the base layer of the MLC image.

rates. However, the decision whether to apply increased protection for the lower layers (because of the resultant overhead), also depends on the effectiveness of other available error concealment techniques. For example, for video transmission where temporal replacement schemes based on motion compensation tend to be effective in concealing errors, the impact of MLC on image quality improvement is not considerable. However, in the case of fast browsing of uncorrelated images in real-time, where only spatial error concealment is possible, MLC with adequate protection can be a viable option.

VI. Acknowledgements

The authors' research in these areas is being supported by a DDR&E Focused Research Initiative under ARO Grant No. DAAH 04-95-1-0248.

Appendix A: The Mechanics of Multi-Resolution Layered Coding

We denote the number of layers used (at different spatial resolutions) by N and the layers as $L_0, L_1, ..., L_{N-1}$. L_0 is the non-differentially coded base layer and L_{N-1} is the finest enhancement layer leading to a full-resolution image. Assuming a reduction in resolution by a factor of two ($r = 2$) in both the horizontal and vertical dimensions and the extreme case where the base layer is reduced to a single block, the (maximum) number of layers, N is given by

$$N = log_r(min(\lfloor R_h/B_h \rfloor, \lfloor R_v/B_v \rfloor)) + 1 \ .$$

where $R_{h,v}$, $B_{h,v}$ are the horizontal and vertical resolutions and the horizontal and vertical block sizes, respectively. For example, a 256×256 image using 8×8 pixel blocks requires 6 layers to be layered in full.

Now consider a MLC image with a resolution of (W, H). Each layer in the image pyramid is recursively generated by convoluting the original image L with the interpolation kernel $h(m,n)$ [8]. The image is first decimated and reduced in resolution by a factor of r in each dimension through downsampling. With $0 \le l < N$, $0 \le i < W_l$, $0 \le j < H_l$, a reduced Gaussian image g_l at level l in the image pyramid is generated as

$$g_l(i,j) = \sum_{m=-M}^{M} \sum_{n=-M}^{M} h(m,n) g_{l+1}(ir+m, jr+n) \ , \quad (1)$$

where a symmetric window of $(2M+1) \times (2M+1)$ is used. With the weight parameter a $= 0.5$ used in our implementation, the weighting function $h(m,n)$ has a triangular shape rather than a Gaussian-like and results in a 3×3 filter window (i.e., $M = 1$). Each differential layer L is generated by subtracting the upsampled image from the downsampled image at each corresponding level, leading to a Laplacian pyramid. The upsampled image $\tilde{g}_l$ at level l, which consists of spatially interpolated values from its preceding layer g_{l-1}, increases in resolution by a factor of r in both horizontal and vertical directions and is generated as

$$\tilde{g}_l(i,j) = r^2 \sum_{m=-M}^{M} \sum_{n=-M}^{M} h(m,n) g_{l-1}\left(\frac{i-m}{r}, \frac{j-n}{r}\right), \ 0 < l < N \quad (2)$$

where $h(m,n)$ is the same smoothing filter as in (1) and non-integer tuples of $(i-m)/r$ and $(i-n)/r$ are excluded from the calculation.

To prevent the propagation of the quantization error from a layer, an error feedback loop can be used as shown in Fig. 1. This feedback loop, adopted in the HJPEG coder, is identical to the one in the pyramid with quantization error feedback analyzed in [14] and can prevent the error from being accumulated to subsequent layers. Since the quantization error at each level is corrected at the next level, propagation of the error (and rounding off error from DCT) is bounded by that of the last layer L_{N-1} and can possibly be eliminated if the last layer is losslessly corrected.[8] Considering the quantization error feedback, g_{l-1} in (2) is then expressed as

$$g_{l-1} = \tilde{g}_{l-1} + L_{l-1} \ .$$

where $\tilde{g}_{l-1}$ is an upsampled prediction image from the previous $l-2$ layers and L_{l-1} is the differential layer at level $l-1$. Hence, each differential layer $L_1, L_2, ...L_{N-1}$ is recursively obtained as

$$L_l = g_l - \tilde{g}_l, \quad 0 \le l < N \ .$$

Note that $L_0 = g_0$ as $\tilde{g}_0 = 0$. Therefore, a full-resolution image can be reconstructed by summing all differential layers to the base layer as

$$g_{r,N-1} = \tilde{g}_{N-1} + L_{N-1} \ . \tag{3}$$

Turning to representation of errors now, suppose D_l represents the lost blocks at level l. Then, the affected region from the loss at the layer is defined as

$$R_l = \sum_{(i,j)\epsilon D_l} L_l(i,j), \quad 0 \le l < N \ .$$

Let a reconstructed image in the presence of block loss at level l be δ_l, defined as $\delta_l = \tilde{g}_l + L_l - R_l$ for $0 \le l < N$, in the presence of block loss. From (3), the reconstructed error image at full-resolution $g_{e,N-1}$ is then

$$g_{e,N-1} = \delta_{N-1} \ .$$

Note that the residual error is local within each layer and its propagation over the layers L_{l+1}, L_{l+2}, ... , L_{N-1} is non-overlapping. This bounded residual error facilitates the use of previous layers for concealment in case of block loss. This compensates for errors rather gracefully as the image quality within a concealed block is invariant. However, it might entail ringing artifacts and blurred edges when erred blocks are stretched over multiple layers under a high channel error rate. Although a degraded image quality is inevitable, error concealment through interpolation from lower layers might be preferable when $|g_{r,N-1} - g_{e,N-1}| < |g_{r,N-1} - \vec{g}_r|$ is realized where $\vec{g}_r$ is a reconstructed image after available spatial and/or temporal concealment.

[8] In practice, different IDCT implementations render perfect lossless coding impractical. However, the precision of the differential input can be reduced by point transform, which restricts coding-decoding difference at a significantly lower cost [9].

References

[1] D. Ferrari, "Client Requirements for Real-Time Communication Services," *IEEE Communications Magazine*, Vol. 28(11), pp. 65-72, November 1990.

[2] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, pp. 771-781, June 1989.

[3] G. Karlsson & M. Vetterli, "Packet Video and its Integration into the Network Architecture," *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 7, pp. 739-751, June. 1989.

[4] A. Lippman & W. Buttera, "Coding Image Sequences for Interactive Retrieval," *Communications of the ACM*, Vol. 32(7), pp. 852-860, July 1989.

[5] D.S. Lee & K. H. Tzou, "Hierarchical DCT Coding of HDTV for ATM Networks," *Proc. ICAASP'90*, Albuquerque, NM April 3-6, 1990, pp. 2249-2252

[6] N. Shacham, "Multipoint communication by hierarchically encoded data," Proc. *IEEE INFOCOM'92*, pp. 2107-2114, May 1992.

[7] W. Verbiest, L. Pinnoo & B. Voeten, "The Impact of the ATM Concept on Video Coding," *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, pp. 1623-1632, Dec. 1988.

[8] P. J. Burt & E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, Vol. 31, No. 4, pp. 532-540, April 1983.

[9] W.B. Pennebaker & J.L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, 1993.

[10] J. K. Han & G.C. Polyzos, "Networking Applications of the Hierarchical Mode of the JPEG Standard," Proc. IEEE International Phoenix Conference on Computers and Communications (IPCCC'96), Phoenix, AZ, March 1996.

[11] P.N. Tudor, "MPEG-2 video compression," *Electronics & Communication Engineering Journal*, Vol. 7(6), pp. 257-64, 1995.

[12] M. Ghanbari & V. Seferidis, "Cell-loss concealment in ATM video codecs," *IEEE Transactions on Circuits and Systems For Video Technology,* Vol. 3, No. 3, pp. 238-247, June 1993.

[13] L. Kieu & K. Ngan, "Cell-loss concealment techniques for layered video codecs in an ATM Network," *IEEE Transactions on Image Processing,* Vol. 3, No. 5, pp. 666-677, Sept. 1994.

[14] M. Vetterli & K. Uz, "Multiresolution coding techniques for digital video: A review," *Multidimensional Systems and Signal Processing,* Vol. 3, pp. 167-187, Mar. 1992.

[15] P. Pancha & M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission," *IEEE Communications Magazine*, Vol. 32(5), pp. 54-66, 1994.

[16] W.H. Leung, T.J. Baumgartner, Y.H. Hwang, M.J. Morgan, & S.-C. Tu, "A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 8(3), pp. 380-390, April 1990.

[17] G. Blakowski & R. Steinmetz, "A media synchronization survey: Reference model, specification, and case studies," *IEEE Journal on Selected Areas in Communications*, Vol. 14(1), pp. 5-35, 1996.

[18] A.E. Eckberg, "B-ISDN/ATM Traffic and Congestion Control," *IEEE Network*, Vol. 6(5), pp. 28-37, September 1992.

[19] D.M. Lucantoni & S.P. Parekh, "Selective Cell Discard Mechanisms for a B-ISDN Congestion Control Architecture," 7th International Teletraffic Congress Symposium, 1990.

[20] J. Kurose, "Open Issues and Challenges in Providing QoS Guarantees in High-Speed Networks," *Computer communications review*, Vol. 23(1), pp. 6-15, 1993.

[21] J.C. Pasquale, G.C. Polyzos, & G. Xylomenos, "The Multimedia Multicast Problem," *Multimedia Systems*, ACM/Springer-Verlag (in press).

[22] E.W. Biersack, "Performance Evaluation of Forward Error Correction in ATM Networks," Proc. ACM SIGCOMM'92, pp. 248-257, August 1992.

[23] A. Albanese, J. Blomer, J. Edmonds, M. Luby, & M. Sudan, "Priority Encoding Transmission," Proceedings, 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, pp. 604-612, November 1994.

[24] J.C. Pasquale, G.C. Polyzos, E.W. Anderson, & V.P. Kompella, "The Multimedia Multicast Channel," *Internetworking: Research and Experience*, Vol. 5(4), pp. 151-162, December 1994.

[25] J.C. Pasquale, G.C. Polyzos, E.W. Anderson, & V.P. Kompella, "Filter Propagation in Dissemination Trees: Trading Off Bandwidth and Processing in Continuous Media Networks," Proceedings Fourth International Workshop on Network and Operat-

ing System Support for Digital Audio and Video, pp. 259-268, November 1993.

[26] M. Khansari, A. Jalali, E. Dubois, & P. Mermelstein, "Robust Low Bit-Rate Video Transmission over Wireless Access System," *Proc. ICC*, Vol. 1, pp. 571-575, Sept 1994.

[27] D.E. Comer, *Internetworking with TCP/IP*, Vol. 1, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1995

[28] M. Kawarasaki & B. Jabbari, "B-ISDN Architecture and Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 9, pp. 1405–1415, December 1991.

[29] G.C. Polyzos & K. Taylor, "A Prototype Video Dissemination Application over ATM," Proceedings of IEEE ICC'95, Seattle, WA, pp. 1262-1266, June 1995.

[30] K.C. Claffy, H.-W. Braun & G.C. Polyzos, "A Parameterizable Methodology for Internet Traffic Flow Profiling," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1481-1494, October 1995.

[31] C. Huitema, *IPv6: The New Internet Protocol*, Prentice Hall PTR, Upper Saddle River, NJ, 1996.

[32] L. Zhang, S. Deering, D. Estrin, S. Shenker, & D. Zappala "RSVP: A New Resource Reservation Protocol," *IEEE Network*, vol. 7, no. 5, pp. 8–18, 1993.

[33] D. LeGall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, Vol. 34, No. 4, pp. 305-313, April 1991.

[34] Draft Revision of Recommendation H.261: Video Codec for Audiovisual Services at $p \times 64$ kbits/s, Revised 14 March 1990.

[35] A.S.Tom, C.L. Yeh & F. Chu, "Packet video for cell loss protection using deinterleaving and scrambling," *IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 2857-2860, Toronto, Ont., Canada, May 1991.

[36] Q. Zhu, Y. Wang & L. Shaw, "Coding and cell-loss recovery in DCT-based packet video," *IEEE Transactions on Circuits and Systems For Video Technology,* Vol. 2, No. 3, pp. 248-258, June 1993.

[37] A. N. Netravali & B. G. Haskell, *Digital Pictures: Representation, Compression, and Standards*, 2nd ed., Plenum Press, New York, 1995.

[38] M. Wada, "Selective recovery of video packet loss using error concealment," *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, pp. 807-814, June 1989.

[39] NTSC video sequence from Berkeley Abekas video

[40] ftp://ipl.rpi.edu/pub/image/sequence/salesman/gray