Networking Applications of the Hierarchical Mode of the JPEG Standard*

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

Hierarchical coding of images and continuous media can be used to effectively control congestion in high-speed networks supporting interactive multimedia communications. However, the tradeoffs of the use of hierarchical coding have not yet been adequately investigated. We have undertaken an empirical study to investigate the effectiveness of hierarchical coding through the hierarchical mode of JPEG from a network perspective. A static analysis of hierarchical JPEG images in comparison to baseline JPEG images in terms of QoS management by packet discarding is provided in this paper. We also share our experiences in the implementation of hierarchical JPEG.

I. Introduction

Advances in computing and high-speed network technologies have led to the proliferation of multimedia applications. Many such applications generate Continuous Media (CM), such as audio and video, and produce heavy volumes of bursty network traffic. In addition, interactive real-time applications, e.g., teleconferencing, introduce strict delay constraints.

Hierarchical Coding (HC), also referred to as layered coding, techniques split signals into components of varying importance [1], [2]. The aggregation of these components reconstructs the original data, but subsets of the components can also provide various degrees of approximation to the original signal. HC is very important for the efficient use of high-speed packet-switching networks such as the emerging Broadband Integrated Services Digital Network (B-ISDN) based on the Asynchronous Transfer Mode (ATM) standards.

The main issue for such network architectures is the significant congestion control problems that can arise due to the statistical multiplexing of highly bursty signals. A key property of any congestion control approach, which 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. With HC 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. Since it is conjectured that CM will constitute the bulk of future network traffic, this technique can be very effective as a last resort for congestion control.

HC can also play an important role at the receiver because it provides the system software with the capability of allocating resources based on local specifications and priorities. This might, for example, entail deciding to gracefully and dynamically degrade the quality of the received and played-back signal when resources are limited.

There are, of course, tradeoffs between the benefits derived from the relative independence of the components produced by HC and the total volume of data generated by compressing components individually (in order to achieve their independence) and the complexity of the (de)compression.

While the advantages of using HC of images and CM for transmission over packet-switching networks are evident, adequate experience to substantiate its efficacy is not widely available. Even worse, a significant penalty is paid for not conforming to traditional, supported formats [13], [10]. Thus, it is important to contrast the benefits of hierarchical coding against the expected cost of its adoption.

In this paper we attempt a basic cost-benefit analysis focusing on a specific HC technique: the hierarchical mode of the JPEG (Joint Photographic Experts Group) still image compression standard. We have implemented the standard in software and are using it to conduct experiments to evaluate the use of HC in various networking applications. In this paper we present measurements of the size of images after compression and the time to compress and decompress using the hierarchical mode of JPEG (HJPEG) and compare them to similar metrics for the baseline JPEG mode (BJPEG). Our measurements indicate that the overhead resulting from the multiple layers of HJPEG may be insignificant. The use of HJPEG in various networking applications is then discussed in view of these results.

The remainder of this paper is structured as follows. A brief description of the hierarchical mode of the JPEG standard and our implementation of it are provided in Section II. In Section III, we present a static analysis

*This research was supported in part by Orincon Corporation and the UC MICRO program and by ARO FRI grant DAAH049510248.

of HJPEG and compare it to BJPEG emphasizing the network perspective. We discuss networking applications of HJPEG in Section V and present our conclusions in Section VI.

II. The JPEG Still Image Compression Standard

A. Modes of Operation

The JPEG (Joint Photographic Experts Group) standard was developed under the auspices of ISO (ISO 10918-1 JPEG Draft International Standard) and CCITT (CCITT Recommendation T.81), and supports both lossy and lossless compression. The lossy methods are based on the Discrete Cosine Transform (DCT). The standard specifies four modes of operation: sequential, lossless, progressive, and hierarchical encoding [14]. The progressive and hierarchical modes allow for decompression of a partially received signal. Even though this standard was developed with still images in mind, it is also used for video transmission by providing intraframe compression only (often referred to as motion JPEG). Even though intraframe compression provides much lower compression ratios for video than combined intra and interframe compression, it has many other advantages, particularly as far as error resiliency is concerned, because of the independence between frames. Therefore, the development of the JPEG standard is considered an important step for multimedia communications.

Sequential encoding is probably the most common mode for most applications. It involves encoding each image component in a single left-to-right top-to-bottom scan. In the baseline encoding algorithm, each component of the source image is divided into 8x8 pixel non-overlapping blocks. The pixel values in each such block are first shifted from unsigned to signed integers and then input to the forward DCT. The resulting 64 DCT coefficient values can be regarded as the relative amount of 2D spatial frequencies contained in the input image. The DC coefficient is a measure of the average value of the 64 image pixels.

The next step is to quantize the DCT coefficients. The purpose of this step is to achieve further compression by quantizing high-frequency components with a larger step size (i.e., more coarsely). This is because high spatial frequencies require less detailed coding. This step discards visually unimportant information and thus makes the approach "lossy." These quantized coefficients are then entropy encoded, with the DC coefficients being treated specially. Since the DC coefficients are a measure of the average value of the pixels in the block, they are expected to show less variation within the same component, and therefore, are differentially encoded.

The quantized AC coefficients are ordered in a zig-zag sequence starting from the top-left corner and traversing the nearest cells first. This ordering puts the low-frequency coefficients before the high-frequency ones, and thus facilitates entropy coding. The coefficients are first run-length encoded and then coded using Huffman or arithmetic coding. The output from the entropy encoder is the output of the JPEG encoder. The JPEG decoder simply reverse this process, using an entropy decoder, dequantizer, and the inverse DCT to reconstruct the image.

B. The Hierarchical Mode of the JPEG Standard

The hierarchical encoding mode (HJPEG) encodes an image at multiple resolutions, each differing from its adjacent level by a factor of two in the horizontal or vertical directions, or both. This is similar to the pyramid decomposition technique presented in [4]. The image is first bandsplit on the basis of spatial frequency and subsampled by the desired number of multiples of 2 in either or both dimensions. For DCT-based HC this new reduced size image (called a frame in JPEG terminology) is encoded using the sequential mode described previously. Then, the encoded reduced-size image is decoded, interpolated and upsampled by 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. The difference image (called a differential frame) is then encoded using the sequential mode. Finally, the last two steps are repeated until the original image at full resolution has been encoded.

We have implemented the sequential DCT-based HJPEG[1] using the existing baseline JPEG code from the IJPEG group. HJPEG differs from the non-hierarchical mode of JPEG in its coding model (i.e., the procedures used to convert input data into symbols to be coded) which uses reference images for the reconstruction of differential frames. For non-differential frames of HJPEG the coding model is identical to that of the sequential mode. However, the coding model for differential frames was modified to handle the signed two's complement differences. The DPCM procedure for coding the DC coefficients was also modified for differential frames, since all coefficients in the layers are differentially coded. The use of full-frame buffers for the reference image in upsampling/downsampling, speeded up the image reconstruction process for differential frames. Modified triangular filters used for both upsampler and downsampler in the implementation seem to produce a better compression rate than a simple bi-linear interpolation from the adjacent pixel values provided as a sample in the JPEG specification. For brevity, details of DCT-based coding and the implementation of HJPEG are omitted in this paper. However the process of constructing differential frames and reconstructing the image in HJPEG is depicted in Fig. 1.

III. Experience with the Hierarchical Mode

A. HJPEG Compression Measurements

Hierarchical coding (HC) is known to provide lower compression ratios than non-HC at the same image quality. HJPEG is not an exception. One of the sources of increased overhead is the process of upsampling and down-

[1] Implementations of HJPEG are hardly available, if not nonexistent. This is partly because of the wide use of the baseline mode of JPEG (which is sophisticated enough for most applications) and partly because of the complexity involved in the implementation of HJPEG.

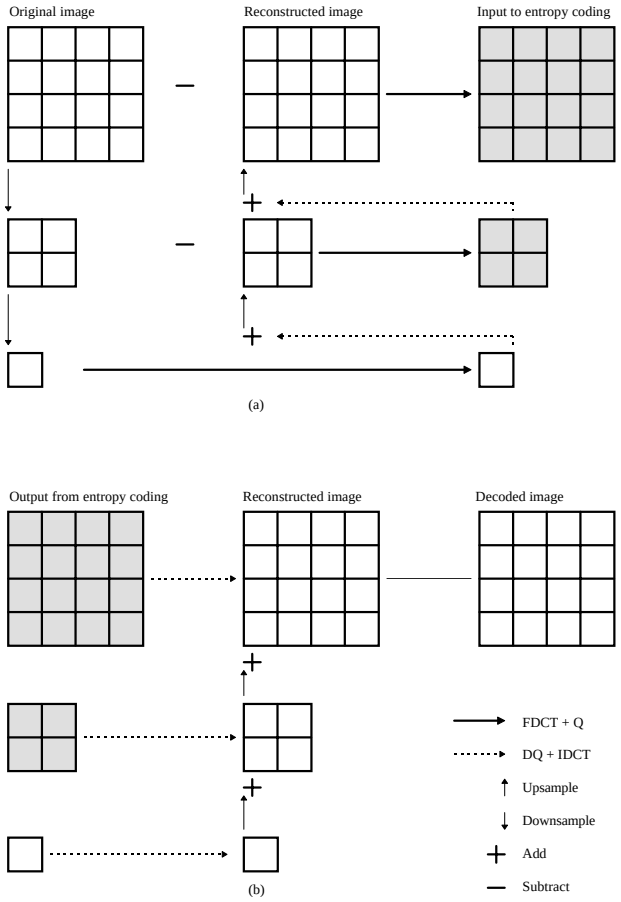

Fig. 1. (a) Coding and (b) decoding procedure of DCT-based differential frames, exemplifying 3-layer HJPEG. Each layer differs in resolution from its adjacent layer by a factor of two in both horizontal and vertical dimension. Each shaded one represents a differential frame (blocks).

sampling within the intraframe compression process. An inefficient sampling filter can cause further error corrections in the following frames and lead to a lower compression ratio.

To get a better understanding of the overhead of HC, 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 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). These figures show far better performance than expected, considering the initial results[2] reported in [5].

[2] In [5], comparison was made between a 3-layer HJPEG image (downsampled by 2 in each direction for each frame) and the progressive mode of JPEG. The HJPEG image has about 33% overhead over the image coded with the progressive mode.

We have observed that downsampling of reasonable size images (from 200×150 and up) by 2 both horizontally and vertically yielded almost comparable image quality as the full-resolution image (Fig. 2). Note that the total size, up to and including the third frame of the 4-layer HJPEG image, constitutes only 35% of the BJPEG size and takes 43% of the time to decode compared to the BJPEG (Fig. 3 and 4). 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.

As discussed previously, the second data set comprises a set of images (video frames). We have performed the same analysis to these images, 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 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.

Fig. 5 demonstrates the low and rather constant bit rate of the base signal while the enhancement layer exhibits significant fluctuations of bit rate. In case of the enhancement layer, the peak-to-mean ratio is slightly lower than that of BJPEG. Hence, these results suggest that the effectiveness of HC results mainly from the low bit rate of base layer rather than any reduced burstiness of the separated layers. Since the bit rate of the base layer is far lower than that of enhancement layer, applications using HC will improve the efficiency of the network by reserving resources at much lower levels.

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. Note that this observation can be used to provide more economical storage for images. E.g., a major portion of an image, the one providing the high-resolution components, can be stored separately from the basic image (i.e., the low-resolution components). This issue is further discussed in Section V-B. Note that although the file size and the decompression time increase slightly as the total number of layers (frames) used in the HC scheme (i.e., n) increase (e.g., see Fig. 3, 4, and 5), 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 frames of HJPEG with more layers attribute to a slightly larger size (Fig. 3)

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

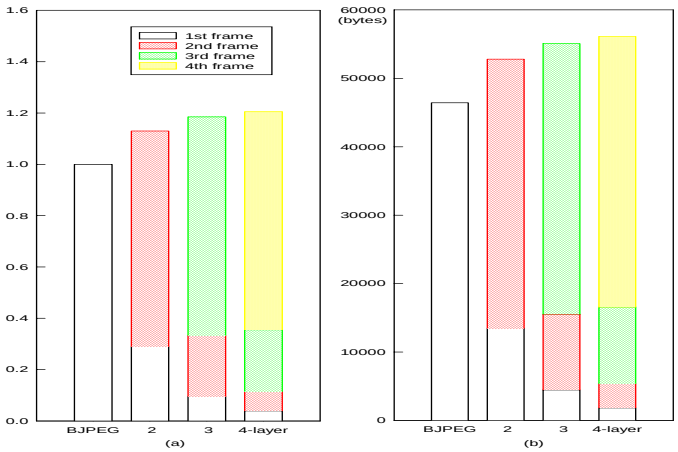

Fig. 3. (a) Ratio of HJPEG frame sizes to BJPEG size (b) frame size of layers in n-layer HJPEG image (bridge).

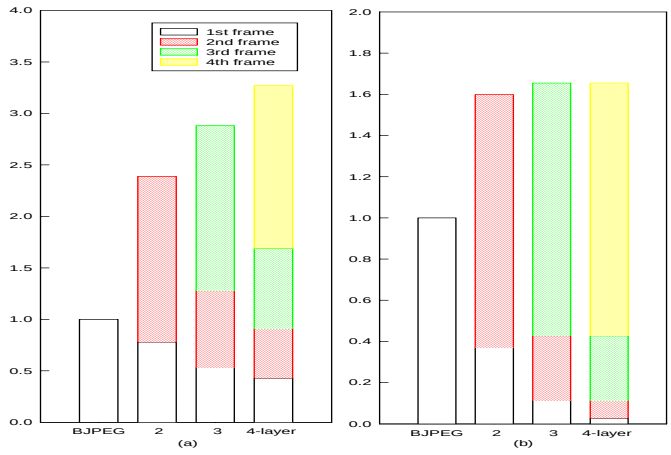

Fig. 4. Ratio of time to (a) compress and (b) decompress n-layer HJPEG image (bridge).

without leading to a better perceptual quality than that of HJPEG with less layers. This suggests that the number of layers (frames) should be decided according to the priority scheme supported by the underlying networks, if the control of finer granularity of image resolution is not desired.

IV. Effect of Cell Loss

Another important issue for interactive real-time applications over ATM and other packet-switching networks is their behavior when cell or packet loss occurs. Although noise-inflicted errors are minimal in the case of transport over fiber-optic media, these high-speed networks still experience cell loss from buffer overflow. Excessive queueing delay in the switching node is also treated as cell loss for applications with strict delay bounds. On the other hand, in the case of wireless networks, where compression is critical, relatively high channel error rates are the norm.

We conducted preliminary experiments to investigate the effect of cell loss without added error concealment techniques and with the following assumptions. The transmis-

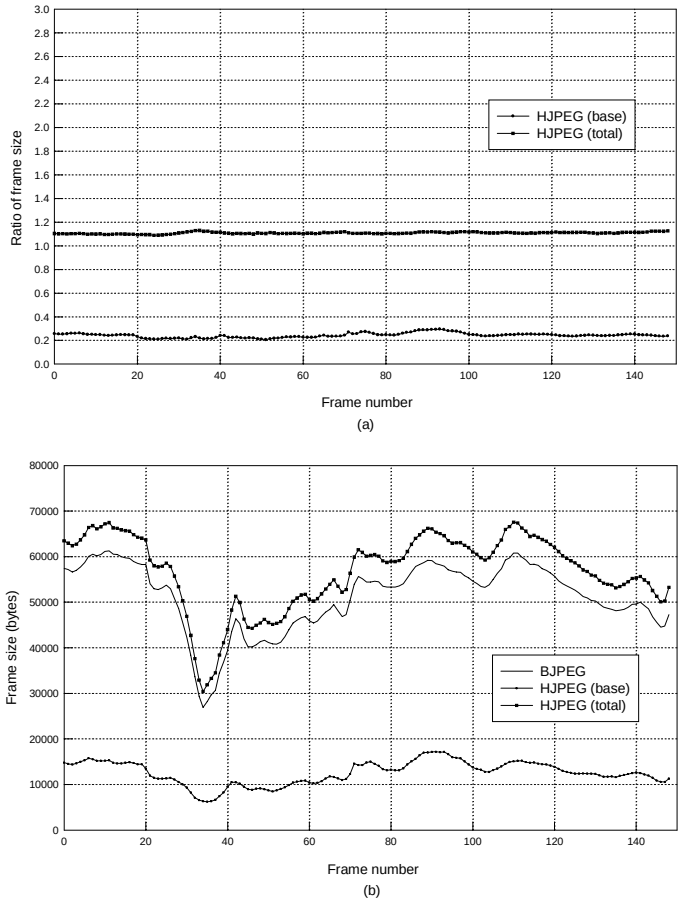

Fig. 5. (a) Ratio of frame size and (b) frame size of 2-layer HJPEG in comparison to BJPEG in the football video clip. HJPEG (base) denotes images after progression to the base layer only.

sion of an n-layer HJPEG image is carried over a guaranteed performance channel (which suffers no cell loss) for frames up to $n-1$ (and including header information and tables), and over a best effort channel for the last layer (n). For BJPEG we assume that no such differentiation is possible and all image information is carried over the best effort channel; however, header information and tables for BJPEG are assumed transmitted via the guaranteed channel to avoid complete failure of displaying the image.

The simulated packet loss was uniformly random with a preset probability. The range of missing packets was confined to the blocks for coefficients after the *start of scan* (SOS) header for both the BJPEG and the last frame of HJPEG. Once a packet is lost, the corresponding 48 payload bytes (as in ATM) are zeroed in the sequence. Only the entropy coded data is assume to become targets for packet loss. No scheme for error concealment was used except for stuffing with zeros for bytes in lost packets. Although zero values are interpreted as neutral grays[3], the decompressed sample values show concentration of dark grays in the upper-left side of each block after the inverse DCT, leading to distortions in the image.

Most compression schemes are sensitive to network errors due to their dependence on differential values (e.g. DPCM). Moreover, when the coding scheme uses run-length coding, the impact resulting mainly from the cell loss affects a wide range of the image and entails unpredictable results. Hence, both BJPEG and HJPEG are extremely susceptible to network errors. However, as it is shown in Fig. 6, HJPEG with a guaranteed base layer has a higher tolerance to errors than BJPEG, which exhibited disastrous effects even from very few cells lost (because of loss of synchronization).

JPEG has an optional provision for error recovery. The RST (restart) marker, a synchronization flag, in the JPEG standard can be used to separate the entropy-coded segments for each interval and allow each segment to be decoded independently of other intervals in the scan [5]. This provision (the actual procedures are not defined by JPEG) can be utilized to resynchronize damaged blocks from packet loss as both encoders and decoders are reinitialized at each interval. Thus, spatial error propagation in DC coefficients (which are differentially coded to reduce bit rate) is limited. Frequent resynchronization with RST markers can help restrict the propagation of errors, but results in an increased processing delay and bit rate.

Although HJPEG with a guaranteed base layer is more tolerant to errors than BJPEG, the quality of the final image of HJPEG in comparison to the image in the absence of loss warrants further investigation into better provisions for concealment and recovery from network errors. Note also that in this case a better strategy seems to discard the lost layer altogether (compare Fig. 6(c)(d) with Fig. 2(c)).

V. Networking Applications of HJPEG

The advantages of hierarchical coding for various applications, e.g., congestion control in high-speed networks, accommodation for heterogeneous end devices, and error resiliency in network transmission have been presented in many papers [2], [12], [6]. In this section, some potential applications of HJPEG are discussed.

A. Multimedia Database Browsing

As multimedia databases, dealing mostly with images, are widely used, content-based information retrieval [16] has gained attention in the database community. This trend of treating images equally with alpha-numeric data will render fast image browsing an important feature in any image database retrieval system. Such systems typically respond to queries with a set of images which are the best matches.

Depending on the resolution of these images and the channel speed, the transmission time and decompression delays for full-resolution can be significant, and might prevent users from browsing the images at their desired pace. With HJPEG, the image database server can be designed to transmit the lower-resolution layers (frames) in browse mode for faster display, and only transmit the remaining finer resolution frames if needed. The latter could actu-

[3] In computing FDCT, the input samples are level shifted to a signed representation by subtracting $2^p - 1$, where p is either 8 or 12 bits for the lossy DCT compression. Hence, these zeros for DCT-based compression are equivalent to neutral grays in DC coefficients.

ally be done without explicit user input, as soon as the transmission of the base layer is complete and as long as the user has not indicated that the image is undesirable.

Finally, HJPEG can also be used for multilayer security, allowing only some of the users to view the images at full resolution, while others can only view low-resolution versions. For example, all users might be given access to a scene, but the faces of people in the scene should not be recognizable for security reasons, except to a subset of the users.

B. Multimedia Database Storage

It is possible to separate each frame of an HJPEG image and store it in a different storage device. For example, the most important frames, essential for initial display can be stored in secondary storage (or even main memory) and the remaining frames can be stored in tertiary storage. As previously shown in Section III-A, the last frame of an image takes a major portion of the total space (approximately 70% of the total file size). Such a scheme has significant economic advantages since images from large image databases (e.g., scientific databases) may need to be retrieved only occasionally and the resolutions of these images tend to be high [9], [13].

C. Multimedia Multicasting

In multicasting, a sender communicates explicitly with a "group," rather than with individual actors who happen to be members of the group. Consequently the sender need not be knowledgeable of group membership, and the group membership may change over time. One interesting class of applications which require group communication includes those with multimedia components, and CM in particular. Efficient multicasting is a fundamental issue for the ultimate success of these multimedia group applications. While in the past multicasting has been viewed as a service of limited use, often provided as an afterthought, this can no longer be the case. A major market for B-ISDN is expected to be selective video distribution, analogous to CATV channels, but where the relatively small number of active receivers and the large number of channels (sources) make broadcast solutions impractical.

HC facilitates multicasting by enabling destinations to adjust the quality of signal they receive each, independently of each other and without the source actually being aware of the adjustment. A similar function can be provided by intelligent network switches at critical points in a network (e.g., at the periphery of a fiber optic ATM network accessing a wireless, lower speed channel) [7]. In [6], an architecture for multicasting CM across heterogeneous networks and to heterogeneous end devices is presented. The multicast routing problem for CM is discussed in [3]. In [11], a first partial implementation of this architecture and some encouraging experimental results are presented. However, the lack of a true HC scheme for video did not allow the development of a practical video dissemination system. The use of our implementation of HJPEG (in a motion JPEG form) would address this issue.

VI. Conclusions

We have presented an analysis of storage requirements and compression and decompression times for images using the hierarchical mode of the JPEG (HJPEG) standard and comparisons to the baseline mode of JPEG (BJPEG). Then we have discussed networking applications of hierarchical coding (HC) and HJPEG in particular.

Our results in Section III-A indicate that the overhead of HJPEG is lower than expected and previously reported and may be negligible, considering the significant potential statistical multiplexing gain. Although slightly lower burstiness of CM can be obtained through HC with intraframe compression, the effectiveness of HJPEG in terms of network efficiency mainly results from the low bit rate of the base layer which should be transmitted over the guaranteed channel.

The bit rate of HJPEG is dependent on the number of layers to a certain extent. Subjective evaluation indicates that the quality of the image including the first $n-1$ layers of an n layer HJPEG image is comparable. These results suggest that the number of layers should correspond to the level of priorities supported by the underlying networks in order to maximize the effectiveness of the scheme.

We also demonstrated the error resilience of HC with guaranteed base layer by investigating the effect of cell loss without added error concealment techniques. The disastrous impact of cell loss in the case of BJPEG indicates that HC may be important for situations where retransmission is impossible or undesirable.

Applications using HJPEG with different priorities can indeed achieve significant performance improvements over BJPEG in terms of faster transmission and reconstruction of partial, but possibly fully functional images. In addition, HJPEG can provide effective and efficient support for multipoint communications to heterogeneous populations and over heterogeneous networks.

References

[1] G. Karlsson and M. Vetterli, "Packet Video and Its Integration into the Network Architecture," *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, pp. 739-751, June 1989.

[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] V.P. Kompella, J.C. Pasquale, and G.C. Polyzos, "Multicast Routing for Multimedia Communication," *IEEE/ACM Transactions on Networking,* Vol. 1, No. 3, pp. 286–292, June 1993.

[4] A. Lippman and W. Buttera, "Coding Image Sequences for Interactive Retrieval," *Communications of the ACM*, Vol. 32, No. 7, pp. 852-860, July 1989.

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

[6] J.C. Pasquale, G.C. Polyzos, E.W. Anderson, and V.P. Kompella, "The Multimedia Multicast Channel," *Internetworking: Research and Experience*, Vol. 5, No. 4, pp. 151-162, December 1994.

[7] J.C. Pasquale, G.C. Polyzos, E.W. Anderson, and V.P. Kompella, "Filter Propagation in Dissemination Trees: Trading Off Bandwidth and Processing in Continuous Media Networks," Proc. *4th International Workshop on Network and Operating System Support for Digital Audio and Video,* November 1993.

[8] J.C. Pasquale, G.C. Polyzos, and V.P. Kompella, "Real-time Dissemination of Continuous Media in Packet-Switched Networks," Proc. *IEEE Compcon Spring '93,* San Francisco, CA, February 1993.

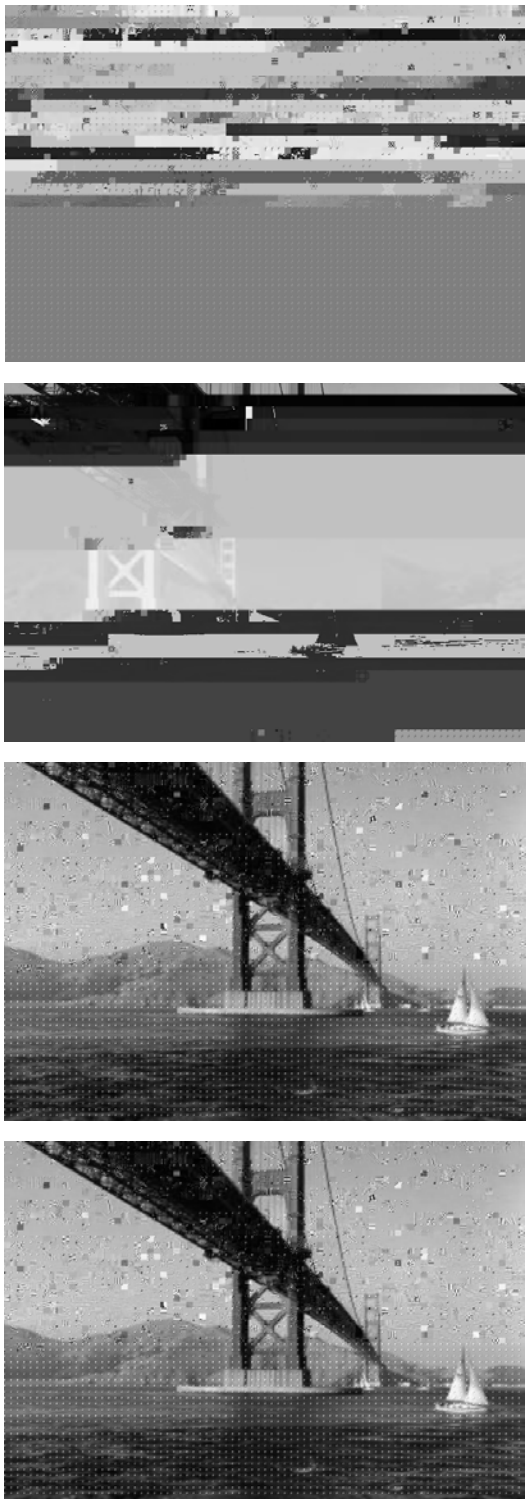

Fig. 6. Effect of cell loss: (a) BJPEG with a 42% cell loss (b) BJPEG with a 2% cell loss (c) HJPEG with a 42% cell loss and (d) HJPEG with a 2% cell loss.

[9] G.C. Polyzos, "Networking for Sequoia 2000," Proc. *4th IEEE International Workshop on Multimedia Communications,* Monterey, CA, April 1992. (Reprinted in *ACM Computer Communications Review,* July 1992.)

[10] G.C. Polyzos, "Image Browsing and Animation over a Local ATM Testbed," Proc. *5th IEEE COMSOC International Workshop on Multimedia Communications,* Kyoto, Japan, pp. S.3.1-6, May 1994.

[11] G.C. Polyzos and K. Taylor, "A Prototype Video Dissemination Application over ATM," Proc. *IEEE International Conference on Communications (ICC'95),* Seattle, WA, pp. 1262-1266, June 1995.

[12] N. Shacham, "Multipoint Communication by Hierarchically Encoded Data," Proc. *IEEE INFOCOM'92,* pp. 2107-2114, May 1992.

[13] K. Taylor and G.C. Polyzos, "Performance Measurements of a Simple Hierarchically Coded Image Animation over Various Network Testbeds," Technical Report CS93-346, Dept. of Computer Science and Engineering, University of California, San Diego, La Jolla, CA, December 1993. (Also available as Sequoia 2000 Technical Report 93/39, U. C. Berkeley.)

[14] G.K. Wallace, "The JPEG Still Picture Compression Standard," *Communications of the ACM,* Vol. 34, No. 4, pp. 30-44, April 1991.

[15] N. Yin, S-Q. Li and T.E. Stern, "Congestion control for packet voice by selective packet discarding," *IEEE Transactions on Communications,* Vol. 38, No. 5, pp. 674-683, May 1990.

[16] E. Binaghi, I. Gagliardi, R. Schettini, "Indexing and fuzzy logic-based retrieval of color images," Proc. *2nd Working Conference on Visual Database Systems*, pp. 84-97, Budapest, Hungary, October 1991.