Techniques for Resilient Transmission of JPEG Video Streams

E.J. Posnak, **S.P. Gallindo**, **A.P. Stephens**, and **H.M. Vin**

Distributed Multimedia Computing Laboratory
Department of Computer Sciences
University of Texas at Austin
Taylor Hall 2.124
Austin, Texas 78712-1188

E-mail: {ejp,gall,asteph,vin}@cs.utexas.edu, Telephone: (512) 471-9732, Fax: (512) 471-8885

Citation: *Submitted for publication*

This research was supported in part by the National Science Foundation (Research Initiation Award CCR-9409666), NASA, Mitsubishi Electric Research Laboratories (MERL), Sun Microsystems Inc., and the University of Texas at Austin.

Techniques for Resilient Transmission of JPEG Video Streams

E.J. Posnak, **S.P. Gallindo**, **A.P. Stephens**, and **H.M. Vin**

Department of Computer Sciences
University of Texas at Austin
Taylor Hall 2.124
Austin, Texas 78712-1188

E-mail: {ejp,gall,asteph,vin}@cs.utexas.edu, Telephone: (512) 471-9732, Fax: (512) 471-8885

September 4, 1994

Abstract

In this paper, we develop a transmission system, referred to as Dual Stream JPEG (DSJ), that improves the resiliency of JPEG video streams by unifying layered encoding with image scrambling techniques. Whereas layered encoding techniques naturally provide resiliency to isolated loss of low priority image data, scrambling techniques make it possible to distribute burst losses throughout the image. In DSJ, this unification is instantiated by employing the discrete cosine transform (DCT) to convert each 8x8 pixel block of an image into frequency domain, partitioning the resultant blocks into high-priority DC coefficients and low-priority AC coefficients, and then scrambling the transmission of encoded blocks of AC coefficients. Furthermore, DSJ defines an adaptation layer that implements efficient error detection and recovery by including the sequence number and offset (relative to the beginning of the cell payload) of blocks carried in each cell. We have evaluated the performance of DSJ through extensive simulations. We present and analyze our results.

Keywords: JPEG, layered encoding techniques, error control, transport protocols

1 Introduction

1.1 Motivation

Recent advances in computing and communication technologies have made it economically viable to provide on-line access to a variety of multimedia information sources (such as reference books, journals, newspapers, images, video clips, scientific data, etc.) over high speed networks. The realization of such information management systems of the future, however, will require the development of transport protocols that ensure efficient and timely delivery of multimedia information to client sites. An important component of transport protocols for multimedia communication is error control. Most conventional transport protocols are founded on the presumption that all the user traffic must be delivered without any loss, and hence, employ Automatic Repeat reQuest (ARQ) techniques (which involve retransmission of lost or corrupted data) for ensuring reliable communication. However, the inherent redundancy in imagery enables transport protocols to recover from transmission errors introduced by the network without retransmissions. Furthermore, the stringent real-time requirements for multimedia communication render ARQ techniques highly ineffective. The design of error control schemes that are optimized for video communication over high speed networks is the subject matter of this paper.

1.2 Relation to Previous Work

To design retransmission-free transport protocols for digital video, several research projects have begun investigating image encoding techniques that enable client sites to reconstruct the image sequence even when a fraction of the transmitted information is lost or corrupted. Most of these protocols can be broadly classified as employing either *explicit* or *implicit* forward error correction (FEC) techniques.

In the case of explicit FEC, redundant information (e.g., parity bits) is transmitted with the original traffic such that the images can be reconstructed at the receiver even if some of the traffic is lost or corrupted by errors [1]. However, for explicit FEC to be efficient, the amount of redundant information that can be transmitted with the original data must be very small. This is because the additional data traffic yielded by the redundant FEC information increases the overall load, which in turn may worsen the loss rate. Furthermore, the usefulness of FEC diminishes when losses are highly correlated since more redundant data must be sent to correct for the loss.

Implicit FEC, on the other hand, is provided by *layered encoding* and *image scrambling* techniques. Layered encoding is founded on the notion that the output of a source coder can be partitioned into an essential layer (which is critical for conveying basic information content of the image) and enhancement layers (which when added to the the essential layer recreate the signal more fully) [3, 4, 5, 6, 7, 11]. Consequently, transport protocols that employ layered encoding techniques arrange the contents of the essential and enhancement layers into packets of high and low priority, respectively, and then ensure that all the high priority packets are timely delivered to each of the client sites. Since the information contained in high priority packets is sufficient for creating a reasonable approximation of the image at the client sites, such a scheme enables the protocol to tolerate isolated loss of packets containing low priority information. The coupling of layered encoding and prioritized transmission, however, does not provide any mechanisms for recovering from the perceptible artifacts yielded by bursty loss of low priority packets.

To address this limitation, researchers have started examining pre-compression image scrambling techniques in which neighboring pixels are transmitted in separate packets spaced out in time [8]. These techniques dramatically reduce the likelihood of neighboring pixels being lost in a single burst, and

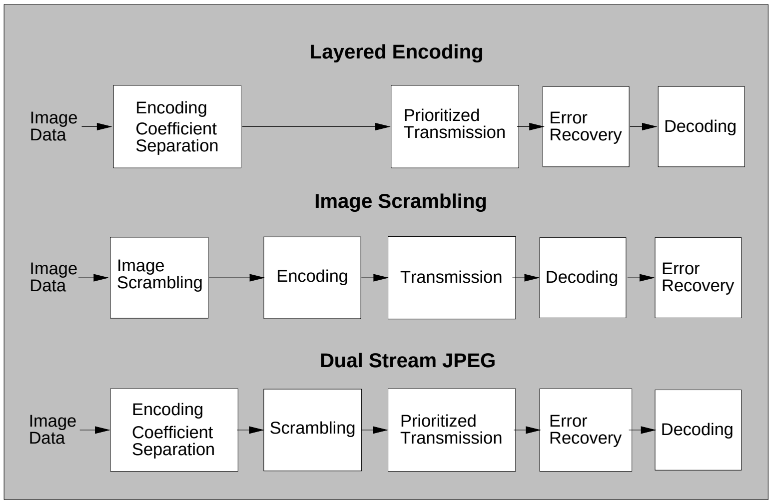

Figure 1 : Comparison of layered encoding and image scrambling techniques with DSJ

thereby make it possible to reconstruct the lost image data by extrapolating from neighboring pixels. Although conceptually elegant, such encoding techniques may adversely affect image compression efficiency. As a result, the bit-rate increase may exacerbate congested conditions and worsen the loss rate.

1.3 Research Contributions of This Paper

In this paper, we develop a transmission system, referred to as Dual Stream JPEG (DSJ), that (1) utilizes layered encoding and prioritized transmission techniques to provide resilience towards isolated loss of media information, (2) implements post-compression scrambling to distribute the effects of bursty loss, and (3) conforms to the JPEG image compression [9] and the ATM communication standards [10]. Specifically, DSJ employs the discrete cosine transform (DCT) to convert each 8x8 pixel block of an image into the frequency domain, partitions the resultant blocks into DC and AC coefficients, and encodes them separately producing high and low priority streams. Furthermore, to achieve a high degree of resiliency towards burst loss, we define a DSJ adaptation layer which implements mechanisms for scrambling the transmission of blocks of AC coefficients, and techniques for detecting and recovering from cell loss. We have evaluated the performance of DSJ through extensive simulations. We present and analyze our results.

The rest of the paper is organized as follows: In Section 2, we describe the layered encoding and scrambling techniques employed in DSJ. The results of simulations are presented in Section 3, and finally, Section 4 concludes the paper.

2 Dual Stream JPEG (DSJ)

Dual Stream JPEG (DSJ) is an integrated encoder and transmission protocol that combines layered encoding and post-compression image scrambling to achieve a high resiliency to bursty cell loss while minimizing the increase in bit rate. In what follows, we first outline the encoding techniques used in DSJ, and then describe an adaptation layer that implements mechanisms for scrambling the transmission of blocks of AC coefficients, as well as error detection and recovery schemes optimized for the transmission of JPEG encoded video streams over ATM networks.

2.1 Layered Encoding and Prioritized Transmission

Since human perception is less sensitive to low frequency components of a video signal, most compression algorithms transform video signals into the frequency domain so as to enable separation into low and high frequency components. For instance, standard encoders, such as JPEG, fragment image data into a sequence of 8x8 pixel blocks which are separately transformed into the frequency domain using the discrete cosine transform (DCT). Each transformed block is then partitioned into the lowest frequency DC coefficient, and a set of high frequency AC coefficients. The DC coefficients of successive blocks are difference encoded independent of the AC coefficients. Within each block, the AC coefficients are scanned in a zig-zag manner to obtain an approximate ordering from lowest to highest frequency, quantized to remove high frequency components, and finally run length and entropy encoded.

DSJ partitions the JPEG encoded video stream into an essential layer (consisting of the DC coefficients) and an enhancement layer (comprising of all the AC coefficients). Whereas the information contained in the essential layer is packetized and transmitted at high priority (which ensures a guaranteed quality of service), information in the enhancement layer is transmitted at a low priority (which provides only a best-effort delivery service). The cell loss priority (CLP) bit in ATM provides a two-level priority mechanism over a single channel which is suitable for this purpose. In the event of congestion, cells which have the CLP bit set will be discarded in favor of those which do not.

With layered encoding and prioritized transmission, each recipient can recover from the loss of AC coefficients by repeating the AC coefficients from the previous block or by interpolating between the neighboring blocks. Clearly, such a recoverability mechanism is founded on the assumption that cell losses are isolated. In practice, however, cell losses yielded by congestion at switches are likely to be bursty. In DSJ, techniques for effectively recovering from burst loss are implemented in an adaptation layer (denoted by DSJ-AL), which is described in the following section.

2.2 DSJ Adaptation Layer (DSJ-AL)

DSJ-AL achieves a high degree of resiliency without an excessive increase in bit-rate by tightly coupling the transmission of video streams with the encoding technique. The two key components of DSJ-AL are: (1) mechanisms for scrambling the transmission of blocks of AC coefficients, and (2) techniques for detecting and recovering from cell loss.

2.2.1 Block Scrambling

To minimize the effect of bursty cell loss, the DSJ-AL orders the transmission of blocks such that cells containing the AC coefficients of adjacent blocks are widely separated in time. To do so, DSJ-AL defines a scalar parameter (denoted by $\textsc{step}$) that, for an image containing $\textsc{ncol} \times \textsc{nrow}$ blocks, uniquely determines the transmission sequence. Specifically, starting from the block located at $(1, 1)$, the transmission sequence can be defined as follows (see Figure 2): If (x_i, y_i) denotes the most recent block of AC-coefficients included in the transmission sequence, then the next block (denoted by (x_{i+1}, y_{i+1})) in the sequence is given by:

$$x_{i+1} = (x_i + \textsc{step}) \bmod \textsc{ncol} \tag{1}$$

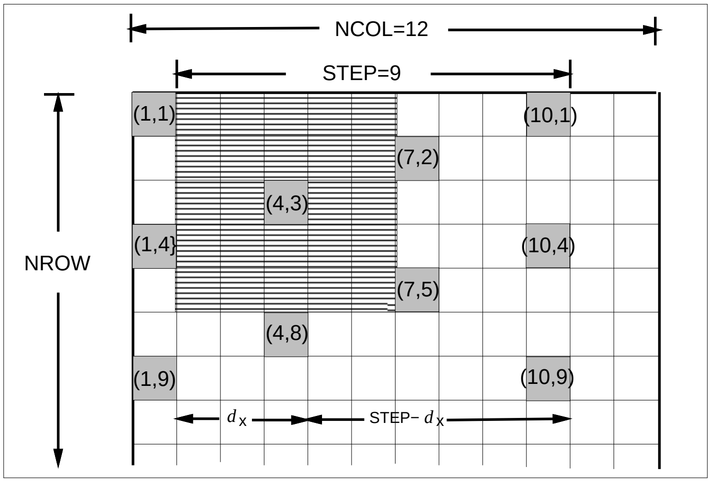

Figure 2 : An image containing NROW x NCOL blocks. The shaded squares represent blocks in a scrambled transmission sequence. A 2-neighborhood of block $(4, 3)$ is highlighted.

and

$$y_{i+1} = \left(y_i + \left\lfloor \frac{x_i + \text{STEP}}{\text{NCOL}} \right\rfloor \right) \tag{2}$$

The sequence continues until the value of y_{i+1} derived using Equation (2) exceeds NROW (i.e., the number of rows in the block array), at which point a new sequence begins with block $(1, 2)$, and so on. Since successive blocks in each subsequence are separated by STEP blocks, the above process is repeated STEP times. The transmission sequence for the entire image is then derived by concatenating each of the subsequences in order.

To maximize the potential for recovery from burst loss, the value of STEP should be chosen such that the transmission of neighboring blocks is separated by as many as blocks as possible. To derive a value of STEP that satisfies this requirement, we will first formally define the concept of *neighborhood* of a block, then derive conditions that help determine the value of STEP.

Definition 1 *For each block (x_i, y_i), its n-neighborhood (denoted by $\mathcal{N}_n(x_i, y_i)$) is defined as the set:*

$$\mathcal{N}_n(x_i, y_i) = \{(x_j, y_j) | \, (|x_i - x_j| \leq n) \wedge (|y_i - y_j| \leq n)\} \tag{3}$$

Specifically, the n-neighborhood of a block denotes the region of $(n+1) \times (n+1)$ blocks surrounding it. For instance, in Figure 2, the shaded region surrounding block $(4, 3)$ indicates its 2-neighborhood.

To ensure that the transmission of neighboring blocks is widely separated, the value of STEP should be chosen such that none of the transmission subsequences derived using Equations (1) and (2) contain blocks which are in the n-neighborhood of each other. Formally, $\forall i, j$, such that (x_i, y_i) and (x_j, y_j) denote the blocks included in a transmission subsequence, the subsequence is said to be *acceptable* if and only if:

$$\forall i, j, i \neq j : \quad (x_i, y_i) \notin \mathcal{N}_n(x_j, y_j) \tag{4}$$

Thus, for all the blocks (x_i, y_i) and (x_j, y_j) in a transmission subsequence, either $(|x_i - x_j|) > n$, or $(|y_i - y_j|) > n$, or both.

To precisely compute the value of STEP, let us assume that block (x_i, y_i) has already been included in a subsequence. Since the n-neighborhood relation is reflexive, to evaluate whether or not a selected value of STEP yields an acceptable subsequence, it is sufficient to only consider the blocks which are added to the subsequence after (x_i, y_i).

Clearly, to ensure that the next block in the subsequence (namely, (x_{i+1}, y_{i+1})) does not belong to the n-neighborhood of (x_i, y_i), the value of STEP should be at least as large as n. Furthermore, none of the blocks located in rows higher than $y_i + n$ can possibly be in the n-neighborhood of block (x_i, y_i). Hence, assuming that $\text{STEP} > n$, to determine whether or not any of the blocks added to the subsequence belong to the n-neighborhood of (x_i, y_i), let us consider all of the blocks chosen from n rows following the row containing (x_i, y_i). Without loss of any generality, let us assume that (x_{i+k}, y_{i+k}) and (x_{i+k+1}, y_{i+k+1}) denote blocks that belong a row that is r rows ($1 \leq r \leq n$) below the row containing (x_i, y_i). Thus, $y_{i+k} - y_i = y_{i+k+1} - y_i = r$. Furthermore, let us assume that $x_{i+k} < x_i < x_{i+k+1}$ (see Figure 2).

In such a scenario, the subsequence is considered acceptable only if:

$$x_i - x_{i+k} > n \;\; and \;\; x_{i+k+1} - x_i > n \tag{5}$$

Since blocks (x_{i+k}, y_{i+k}) and (x_{i+k+1}, y_{i+k+1}) denote successive blocks in the transmission subsequence, we get:

$$x_{i+k+1} - x_{i+k} = \text{STEP}$$

Thus, if we denote $(x_i - x_{i+k})$ as d_x, then the difference $(x_{i+k+1} - x_i)$ is given by $(\text{STEP} - d_x)$. Hence, Condition (5) reduces to:

$$\begin{aligned} d_x > n \;\; and \;\; \text{STEP} - d_x > n \\ \Rightarrow n < d_x < \text{STEP} - n \end{aligned} \tag{6}$$

Furthermore, since $y_{i+k} - y_i = r$, and since successive blocks in a transmission subsequence are separated by STEP, the value of d_x can be derived as:

$$d_x = r * \text{NCOL} \bmod \text{STEP} \tag{7}$$

Thus, substituting the value of d_x in Equations (6), we get:

$$n < r * \text{NCOL} \bmod \text{STEP} < \text{STEP} - n, \qquad 0 < r \leq n \tag{8}$$

Consequently, given the values of NCOL, NROW, and n, if a value of STEP is determined such that Condition (8) satisfied, the corresponding subsequences are guaranteed to be acceptable.

To enable DSJ-AL to efficiently compute the value of STEP, observe that the smallest value of STEP that may satisfy Condition (8) is $(2n + 2)$. The upper bound on the value of STEP, on the other hand, can be derived by using the information regarding the maximum length of burst loss. Specifically, if the transmission system is to ensure that the artifacts introduced by burst losses of length L are not clustered within any n-neighborhood, the value of STEP should not exceed:

$$\text{STEP} \leq \left\lfloor \frac{\text{NCOL} * \text{NROW}}{L} \right\rfloor \tag{9}$$

Thus, to meet the application requirements, DSJ-AL selects the value of STEP to be the smallest value in the interval $[(2n+2), \frac{\text{NCOL}*\text{NROW}}{L}])$ that satisfies Condition (8).

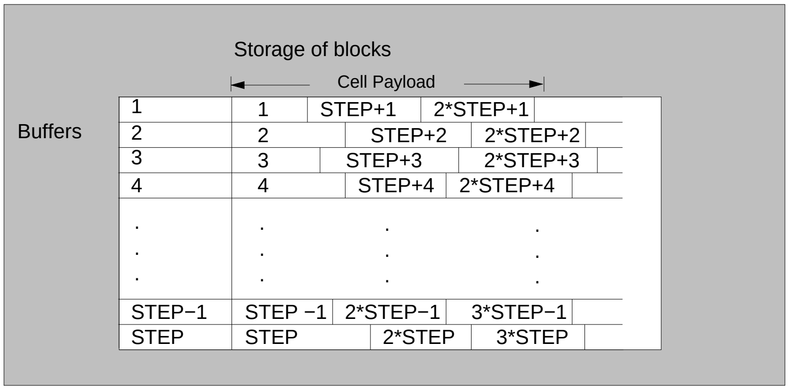

Figure 3 : Array of buffers

To implement scrambling, DSJ-AL maintains an array of STEP buffers to store the entropy encoded blocks of AC coefficients. As the blocks are processed, the DSJ-AL cycles through the array, appending the coded block to the next buffer. To do this, DSJ-AL must delineate blocks of entropy encoded AC coefficients, possibly by sharing the symbol table of a Huffman encoder[1]. Figure 3 depicts the resulting array of buffers. To achieve the desired ordering of transmission, segmentation proceeds by scanning the array horizontally across the rows and filling each cell payload completely.

2.2.2 Detecting and Recovering from Cell Loss

Since the value of STEP uniquely determines the transmission order of blocks of AC coefficients, and since ATM guarantees in-order arrival of cells on a given channel, cell loss can be immediately detected by DSJ-AL at each of the client sites. To facilitate the detection process, DSJ-AL must transmit block sequence numbers in the cell payload, as well as maintain information about the blocks that have already been received. Furthermore, since cell boundaries may not coincide with block boundaries (see Figure 3), resynchronization with the data stream after cell loss will require the DSJ-AL to also transmit the bit offset of the first decodable block within each cell.

At each of the client sites, detection of cell loss requires the DSJ-AL to compare the block numbers received in successive cells. If the sequence of blocks received from the network agree with the sequence defined by the value of STEP, DSJ-AL infers that none of the cells have been lost. On the other hand, a discrepancy between the expected block number and the received block number indicates that all the blocks in the sequence from the expected block up to the received block have been lost. To synchronize with the data stream, DSJ-AL discards all the information in the newly received cell up to the offset marker for the first decodable block. Furthermore, to enable the decoder to successfully decode each 8x8 block, all of the lost AC coefficients can be replaced by zeros or can be interpolated from the neighboring blocks.

---

[1] An efficient implementation may allow the entropy encoding/decoding step to be performed in the DSJ-AL. In this case the encoder passes a stream of difference encoded DC coefficients and a stream of run length encoded AC coefficients to the DSJ-AL. On the receiving end, the DSJ-AL passes the same two streams back to the decoder.

Observe that the block sequence numbers and offset information, together with the value of STEP, are sufficient to detect loss of any cell as long as an encoded AC block does not span more than cells. The inadequacy of the above procedure to handle blocks that span more than 2 cells can be illustrated as follows: Consider an encoded block of AC coefficients (denoted by α) that spans cells i, $(i+1)$, and $(i+2)$. Assuming that block β follows α in the transmission sequence, then the (block number, offset) information contained in DSJ-AL headers for cells i, $(i+1)$, and $(i+2)$ will be (α, o_α), (α, ∞), and (β, o_β), respectively. In such a scenario, the loss of cell $(i+1)$ will not be detected since the client site receives the next block it expects (namely, β) in cell $(i+2)$.

To address this limitation, DSJ-AL includes a sequence number with each cell that enables the client to detect isolated loss. Since the maximum size of a non-encoded block of AC coefficients is 63 bytes (assuming 8-bit coefficients), an encoded block may span at most 3 cells. Thus, a 1-bit sequence number is sufficient. The format of each DSJ cell, which includes a three byte DSJ-AL header (containing a 1-bit sequence number, a 14-bit block number, and a 9-bit offset value) is shown in Figure 4.

2.2.3 Discussion

By tailoring the transmission technique to compression algorithm, DSJ-AL achieves higher resiliency to cell loss than the ATM Adaptation Layer 2 (AAL2), which is intended for use in transmission of digital video over ATM. The limitations of AAL2 mainly stem from its generality, and are briefly outlined below:

- The sequence number in AAL2 enables the detection of loss of up to 64 consecutive cells, but does not provide any mechanism to recover from such loss. Instead, recovery must be performed at the decoder level using a mechanism such as restart markers, which provide coarse-grained recovery at the cost of increased bit-rate. In contrast, the use of block indexing in DSJ-AL permits each client to recover from a burst loss of any length, and immediately resynchronize at the next 8x8 block without requiring any decoder level overhead.

- The length indicator field (LI) is unnecessary since cell payloads, except the last, are completely full. The DSJ-AL does not require a length indicator for the last cell since it can recognize the end of the last block (possibly by using the Huffman code table), and ignores any trailing data in the cell.

- The use of Cyclical Redundancy Check (CRC) per cell is inappropriate for transmission of JPEG images over ATM networks, which are expected to have extremely low bit-error rates. To justify this claim, consider the transmission of a compressed image containing n bits on a link with a bit-error rate B. If we consider the transmission of each bit as a bernoulli trial with probability B of being in error, then the the probability of r bit errors occurring during the transmission of n bits can be determined using the poisson approximation as:

  $$P(r) = \left(\frac{\lambda^r}{r!}\right) e^{-\lambda}$$

  where $\lambda = nB$. Consequently, the probability of transmitting an image containing 10^6 bits with zero errors on a link whose bit-error rate is 10^{-12} can be computed as:

  $$P(0) = e^{-10^{-6}} = 0.999999$$

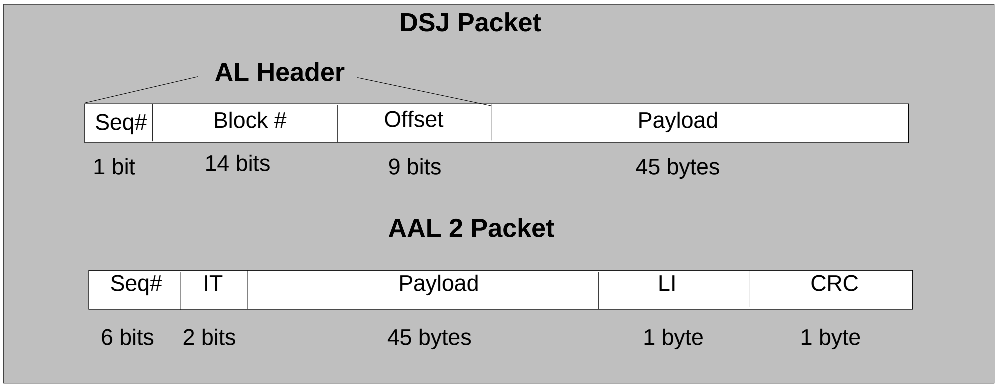

Figure 4 : A comparison of the DSJ and the AAL headers. IT=Information Type, LI=Length Indicator, and CRC=Cyclical Redundancy Check

> Thus, the event of a bit error is highly unlikely, even for large images. Furthermore, only a fraction of bit errors will be catastrophic or even visually noticeable.

3 Experimental Evaluation

The resiliency to cell loss of DSJ was compared to that of the popular JPEG File Interchange Format (JFIF), which uses restart markers to provide recovery from cell loss. To evaluate the effectiveness of these approaches, the CCITT cell discard model [2] was used to simulate bursty cell loss in an ATM network. Five different images, including pictures of an astronaut, the Goldengate bridge, the planet Jupiter, Lena, and a kitten, were used for comparison. Each of the five images was JPEG encoded using restart markers after every 16 macroblocks to enable the decoding in the JFIF case. To isolate the effects of layered encoding and scrambling, we examined DSJ both with and without scrambling. In the following discussion, we will refer to these three transmission formats as JFIF, Layered (i.e., DSJ without scrambling), and DSJ.

For each of the five images 100 simulation runs were made in which the compressed image was packetized in all three formats and subjected to cell loss using the simulator. Whereas all cells in the JFIF format were eligible for discarding, in the case of the Layered and DSJ formats, only cells in the low priority AC stream could be selected. By initializing the simulator with the same random number seed, the same number and sequence of cells were discarded from each of the three cell streams. No recovery mechanisms were employed in the JFIF case, and for the DSJ and Layered formats lost AC coefficients were replaced by zeros.

For each of the five images, the average Signal to Noise Ratio (SNR) of the decoded images over all runs was calculated for JFIF, Layered and DSJ. Figure 5 shows a bar graph comparing the average SNR of the three techniques when the cell loss probability is 10^{-1} and the mean burst length is 16 cells. It illustrates that layered encoding significantly improves the SNR for all images. Since the SNR metric does not quantify the distribution of the errors introduced by cell loss, and is solely based on the number of errors, both Layered and DSJ techniques yield approximately the same values of SNR.

To enable a subjective evaluation, Figure 6 depicts the decoded images for each of the three transmission formats. The pictures illustrate that, whereas the highly visible streaks in the JFIF image

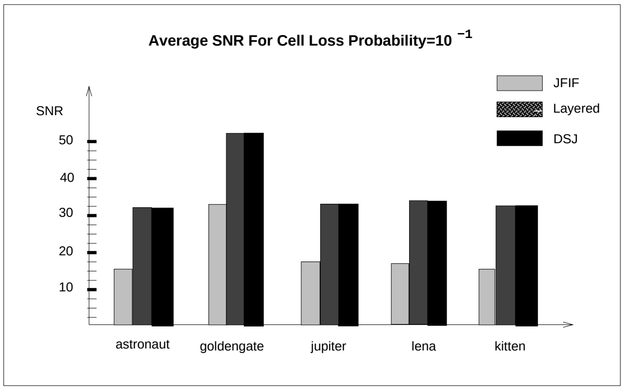

Figure 5 : The average SNR of three transmission techniques (JFIF, DSJ with no scrambling, and DSJ with scrambling) for five different images

are due to the extremely coarse granularity of loss (16 macroblocks) and lack of any error recovery, finer loss granularity (1 8x8 block) and DC coefficient approximation make the DSJ transmitted images much more resilient to cell loss. Furthermore, the block scrambling technique employed in DSJ disperses the blurry streaks (noticeable in the un-scrambled layered encoded image) into isolated errors, and thereby significantly improves the image quality.

4 Concluding Remarks

In this paper, we have shown that by coupling layered encoding with post-compression scrambling techniques, DSJ achieves high resiliency to cell loss while incurring a modest increase in bit rate required for transmission of digital video. The use of block numbers and offsets in the cell header enables DSJ to recover from cell loss at the granularity of an 8x8 block with less than 6% overhead. Our simulations have demonstrated that, even at very high loss rate (e.g., 10%), DSJ yields barely perceptible image degradation. Quantitatively, employing DSJ results in about 100% improvement in SNR over JFIF. Furthermore, post-compression scrambling of AC blocks distributes the effects of bursty loss, and hence, significantly improves the quality of the reconstructed image.

REFERENCES

[1] E.W. Biersack. Performance Evaluation of Forward Error Correction in ATM Networks. In *Proceedings of ACM SIGCOMM'92, Computer Communications Review, Vol. 22, No. 4*, pages 248–257, August 1992.

[2] Experts Group for Video ATM Coding. Working Document AVC-205. In *SGXV Working Party XV/I*, 1992.

[3] M. Ghanbari. Two-Layer Coding of Video Signals for VBR Networks. *IEEE Journal on Selected Areas in Communications*, 7(5):771–781, June 1989.

[4] K. Joseph, J. Zdepski, R. Siracusa, and D. Rayachaudhuri. Design and Performance Evaluation of a Variable Bit-Rate (VBR) Video Compression Algorithm for ATM Networks. In *Proceedings of the IEEE GLOBECOM*, pages 0009–0015, 1991.

Figure 6 : Three decoded images which have been transmitted using (a) JFIF (b) Layered and (c) DSJ formats. Each image has suffered 10% cell loss.

[5] T. Kinoshita, T. Nakahashi, and M. Maruyama. Variable Bit-rate HDTV CODEC with ATM Cell Loss Compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(3):230–236, June 1993.

[6] F. Kishino, K. Manabe, Y. Hayashi, and H. Yasuda. Variable Bit-Rate Coding of Video Signals for ATM Networks. *IEEE Jour. Selected Areas Communications*, 7(5):801–806, June 1989.

[7] V. Seferidis. Video Coding for ATM Networks. *Ph.D. Dissertation, University of Essex,* July 1993.

[8] C.J. Turner and L.L. Peterson. Image Transfer: An End to End Design. In *Proceedings of ACM SIGCOMM'92, Baltimore*, pages 258–268, August 1992.

[9] G.K. Wallace. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4):31–44, April 1991.

[10] CCITT Study Group XVIII. ATM Layer Specification for B-ISDN. In *I.361*, 1992.

[11] Q. Zhu, Y. Wang, and L. Shaw. Coding and Cell Loss Recovery in DCT-Based Packet Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(3):248–258, June 1993.