

Unequal Error Protection of Embedded Multimedia Objects for Packet-Erasure Channels

Madhavi Marka and James E. Fowler

Department of Electrical and Computer Engineering

Engineering Research Center

Mississippi State University, Mississippi State, MS

Abstract—The application of forward-error-correcting codes to data organized as multiple, independent multimedia objects and encoded with modern embedded coders is investigated. Capitalizing on the strict importance-ordering characteristic of embedded encodings, the strength of the error protection is optimized such that data that is more important to the reconstructed quality of the dataset is assigned stronger protection. The focus of the investigation is on providing this optimization while maintaining the ability to independently access the individual multimedia objects. Experimental results are presented for still-image objects that illustrate that the desired independent-access ability comes at a cost in reconstruction quality, and that this cost increases as the channel-loss conditions actually experienced degrade from those for which the optimal protection arrangement was designed.

I. INTRODUCTION

Multimedia scenes are commonly represented as compositions of multiple, distinct objects so as to permit access and manipulation of one object independently of others. For this reason, objects in a multimedia dataset are encoded independently along with information on size and position within the dataset, so that, at the decoder, users can both decode an object of interest and also interactively manipulate object composition without the need to decode the entire scene. The recent MPEG-4 standard relies heavily on object-based representation of scenes, and the standard includes definitions for arbitrarily shaped video objects and arbitrarily shaped still-image texture objects, among others.

Object-based compression can alleviate many problems associated with the storage and transmission of the sizable information associated with multimedia datasets. However, when communication of multimedia data takes place over networks, the multimedia data is divided into packets which are transmitted individually, and some of these packets may be lost in transit across the network due to network failure and congestion. Conventional methods for handling this data loss require identification and retransmission of the lost packets which can cause significant network delays, waste of network bandwidth, and exacerbation of the situation that caused the loss in the first place. Methods that avoid retransmission include error-protection algorithms which assign forward-error-correcting (FEC) codes to

This work was done as part of the High Performance Visualization Center Initiative (HPVCI) funded by the Department of Defense High Performance Computing Modernization Program (HPCMP).

the data. The most straightforward approach to FEC protection is that of equal loss protection (ELP) in which the strength of error protection is applied equally to all portions of the data to be transmitted. Alternatively, unequal-error-protection (UEP) algorithms assign unequal amounts of FEC protection to the data in an effort to vary protection strength according to the importance of various portions of the data. When UEP FEC codes are applied to data that has been subjected to lossy compression, the measure of “importance” in determining the application of the UEP redundancy is usually the quality of the reconstruction. In general terms, the UEP approach tends to perform better than ELP in terms of quality, since one can better optimize the UEP placement so as to maximize the quality of the reconstruction of the data. UEP is particularly suited to embedded coding since the coded bitstream in this case is arranged in order of decreasing importance during the encoding process.

The work presented here investigates the FEC protection of object-based embedded coding. Specifically, we explore the assignment of UEP to objects of a multimedia dataset with the goals that each object 1) is protected within itself according to the importance of each of its bits to the reconstruction quality of the object, 2) has an object-level amount of error protection proportional to the object’s importance to the reconstruction quality of the scene, and 3) can be accessed independently of other objects of the image. Hereafter, only arbitrarily shaped still-image objects, which are a special case of multimedia data, are considered. The general approaches considered, though, apply to embedded codings of other multimedia forms. We note that the current MPEG-4 standard supports embedded coding of still 2D texture objects similar to the image objects considered here.

II. UNEQUAL ERROR PROTECTION (UEP) FOR EMBEDDED BITSTREAMS

UEP is an FEC framework that assigns error-protection codes to bitstreams such that the most important information receives the greatest protection in order to provide graceful degradation of reconstruction quality as packet losses increase. Here, we focus on the algorithm due to Mohr *et al.* [1] which designs a UEP FEC assignment optimizing reconstructed quality for a given probabilistic packet-loss model. In this algorithm, FECs and data bytes form a “stream,” with the number of streams equal

to the number of bytes in each packet to be transmitted. An example of this arrangement is illustrated in Fig. 1(a). Since the bitstream is embedded, earlier parts of the bitstream are more important than latter parts. Thus, the algorithm naturally assigns a greater number of FECs to earlier streams than to latter streams, and optimizes the FEC arrangement to maximize expected quality of the reconstruction subject to the loss model.

The tenet central to the algorithm of [1] is that all the bytes of a stream can be recovered if the number of packets lost is less than or equal to the number of FEC bytes in that stream; such is the case when Reed-Solomon codes are applied to each stream to generate its FEC bytes. For example, if packet 4 in Fig. 1(a) is lost while the other packets are received, the initial 26 bytes of data can be recovered after inverting the FEC code. Additionally, data bytes 27-29, 31, and 32 are received correctly but data byte 30 is lost. In an embedded bitstream, a byte cannot be decoded unless the previous byte is decoded. In this case, bytes 31 and 32 are not useful because byte 30 is lost and thus a total of 29 bytes are used to decode and reconstruct the image. Fig. 1(b) illustrates the data recovery for this case.

The algorithm in [1] is driven by a “quality-vs-prefix” profile, a table which gives the reconstruction quality for every possible prefix of the embedded bitstream. Such a quality-vs-prefix profile is easily generated during encoding with embedded compression algorithms.

III. OBJECT-BASED UNEQUAL ERROR PROTECTION

In an embedded coding, each object is coded as a sequence of enhancement layers. In many embedded coding algorithms, these enhancement layers are bitplanes from wavelet coefficients, but other arrangements (e.g., fractional bitplanes) are possible. Below, we consider two approaches for simultaneously transmitting the enhancement layers from multiple objects. In the first scenario, the layers from multiple objects are arranged in an interleaved fashion and transmitted over a single network connection, or channel. In the second scheme, each object is transmitted simultaneously over its own channel. In the remainder of this manuscript, we will focus on the case in which we have two objects although the discussion is easily generalized to a greater number of objects.

A. Combined Unequal Error Protection (CUEP)

Assume that we have two objects that have been independently embeddedly coded with M_1 layers in object 1 and M_2 layers in object 2, $M_2 \geq M_1$. The individual layers of the objects can be interleaved to form a single compressed bitstream that is transmitted over a single network channel. Specifically, interleaved bitstream B is

$$B = L_{11}, L_{21}, L_{12}, L_{22}, \dots, L_{1M_1}, L_{2M_1}, L_{2(M_1+1)}, \dots, L_{2M_2},$$

where L_{ij} is layer j from object i . In this arrangement, both objects are refined equally fast, and bitstream B forms a representation of the entire scene that is approximately embedded. The bitstream, \tilde{B} , that will be decodable after losing some packets in transmission will be some truncated version of B , while the length of \tilde{B} will depend on the arrangement of the FEC protection that is applied. Specifically, suppose for example, the decodable bitstream is $\tilde{B} = L_{11}, L_{12}, \tilde{L}_{12}$. In this case, we reconstruct the image by 1) reconstructing object 1 using the first layer, L_{11} , and the truncated version of the second layer \tilde{L}_{12} , 2) reconstructing object 2 using the first object-2 layer, L_{21} , and 3) compositing the reconstructed objects 1 and 2 to form a reconstructed image. In our experiments, the quality measure of interest is a whole-image PSNR calculated as the PSNR between the composited reconstructed image and the original image.

The most straightforward approach to providing UEP for the bitstream B would be to apply the UEP algorithm of [1] to it directly. Fig. 2(a) illustrates a packet arrangement typical to this approach. The layers of objects 1 and 2 are interleaved, and the FEC codes are assigned to the individual object bytes according to their contribution to the entire-image PSNR. Unfortunately, this arrangement does not provide independent access to the objects since the individual-object bytes are not in continuous packet locations. That is, any given packet might hold both object-1 and object-2 bytes. Hence, access to individual objects is possible only after the decoder inverts the entire UEP code matrix, which can occur only after the encoder sends all packets, and the decoder determines which packets are missing.

Applying the algorithm of [1] to the interleaved bitstream B would require a “PSNR-vs-prefix” profile for B . Given PSNR-vs-prefix profiles generated while coding the individual objects, a PSNR-vs-prefix profile for the interleaved bitstream can be easily estimated by converting PSNR values to MSE. Specifically, the PSNR for a prefix of B of length $l_1 + l_2$ is

$$d(l_1 + l_2) = 10 \log_{10} \left[\frac{K_1 + K_2}{K_1 10^{-d_1(l_1)/10} + K_2 10^{-d_2(l_2)/10}} \right],$$

where $d_i(l_i)$ is the PSNR-vs-prefix profile for object i , and object i contains K_i pixels.

Let $p(x_i, N)$ be the probability that x_i packets are lost from a total of N packets, and $D(x_i)$ be the whole-image PSNR obtained in this case. The expected PSNR is $\bar{D} = \sum_{i=0}^N p(x_i, N) D(x_i)$. Note that $D(x_i)$ depends on the FEC arrangement of the bitstream B as determined by the UEP algorithm of [1], and the PSNR-vs-prefix profile from above, $d(l)$.

B. Individual Unequal Error Protection (IUEP)

In order to ensure independent access to the individual objects in the bitstream B , an alternative to the previous single-channel transmission would be to apply the algorithm of [1] to each object individually and transmit each object simultaneously over separate, identical channels. That is, we would gen-

erate two bitstreams B_1 and B_2 ,

$$B_1 = L_{11}, L_{12}, L_{13}, \dots, L_{1M_1}$$

$$B_2 = L_{21}, L_{22}, L_{23}, \dots, L_{2M_2}.$$

In this case, we have two PSNR-versus-prefix profiles, one for each object. The algorithm of [1] is applied individually to bitstreams B_1 and B_2 to get the FEC configuration for each. Fig. 2(b) illustrates a packet arrangement for this approach. We see in this example that data bytes of objects 1 and 2 are in consecutive packets, while a given packet contains bytes from only one object. Since each object is transmitted in its own channel, this approach ensures independent access to objects. In this example, the decoder can invert the UEP code matrix for object 1 after only the 3 packets are transmitted and accounted for through channel 1—it does not have to receive any packets from channel 2 to do so. The advantage of this method over CUEP is that it provides independent access without requiring access to all transmitted packets.

Let $p(x_i, N_i)$ be the probability that x_i packets are lost from N_i packets in object i . Then $p(x_1, N_1)p(x_2, N_2)$ is the probability that x_1 packets are lost from object 1 and x_2 packets are lost from object 2. The expected whole-image PSNR is $\bar{D} = \sum_{x_1=0}^{N_1} \sum_{x_2=0}^{N_2} p(x_1, N_1)p(x_2, N_2)D(x_1, x_2)$, where $D(x_1, x_2)$ is the whole-image PSNR when x_1 packets are lost from object 1 and x_2 packets are lost from object 2, a quantity easily obtainable from individual-object PSNR-vs-prefix profiles and the UEP FEC arrangements for the bitstreams.

IV. EXPERIMENTAL RESULTS

All experiments are conducted using the shape-adaptive SPIHT implementation in QccPack [2], which is SPIHT [3] modified with a shape-adaptive wavelet transform in a manner similar to [4]. We use two images which have each been manually partitioned into two objects as shown in Fig. 3. In all experiments, a total bit rate of 0.5 bits per pixel (bpp) is allocated for the entire image, FECs included. The tests are done assuming for each channel an exponential packet-loss model at a mean loss rate of 20%. We assume 47 bytes of data in each packet, as ATM packets have a payload length of 48 bytes of which one byte is required for a packet-sequence number.

The FEC arrangement resulting from applying the algorithm of [1] in the CUEP framework is shown in Fig. 4(a) for lena. In Figs. 4(b) and (c), the FEC arrangements resulting from the IUEP approach are shown for each object from lena. In each case, it is observed that FEC protection strength increases with increasing stream number.

In Fig. 5, the expected PSNR is shown for the CUEP and IUEP approaches under a variety of channel conditions; i.e., for each approach, we design the FEC arrangement assuming channels with an exponential-loss model with mean loss rate of 20% and then determine performance using exponential-loss channels with a loss rate of λ , $0 \leq \lambda < 100\%$. In this manner, we evaluate the performance of the UEP code arrangement for

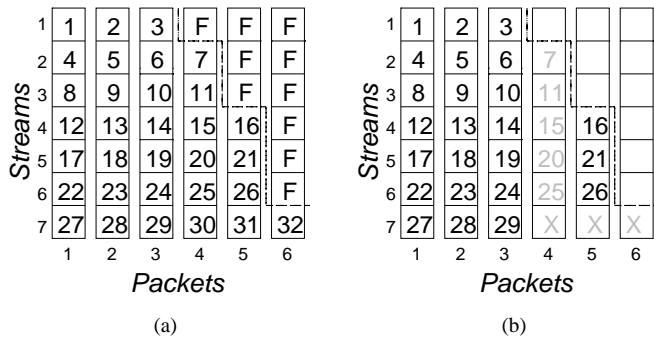


Fig. 1. Example packet arrangement for UEP. (a) Transmitted, (b) data recovery after packet 4 is lost. Data bytes numbered according to their occurrence in the embedded bitstream, F = FEC byte (adapted from [1]).

the situation that the channels encountered are different from those for which we designed the code. In Fig. 5, we see that the PSNR obtained using CUEP is higher than that for IUEP with the difference between the two approaches increasing as the packet-loss rate increases beyond 20%, the amount of loss for which the protection schemes were designed. This difference in PSNR can be considered to be the cost of obtaining independent access to objects, and we see in Fig. 5 that this cost increases as the mismatch between the actual and design channels increases. Actual reconstructed images for lena are shown in Fig. 6.

V. CONCLUSIONS

Two general paradigms for providing object-based UEP have been investigated. These two frameworks tailor FEC codes according to relative importance using embedding encoding and an algorithm for UEP-code assignment. The CUEP approach outperforms the IUEP method in expected PSNR, but does not provide independent access to the objects. Experimental results reveal that there is a cost associated with independent access, and that this cost increases as the mismatch between the actual and design channels increases.

ACKNOWLEDGMENT

Thanks to A. Mohr for helpful discussions and source code.

REFERENCES

- [1] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 819–828, June 2000.
- [2] J. E. Fowler, "QccPack: An open-source software library for quantization, compression, and coding," in *Applications of Digital Image Processing XXIII*, A. G. Tescher, Ed., San Diego, CA, August 2000, Proc. SPIE 4115, pp. 294–301.
- [3] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.
- [4] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrary shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 725–743, August 2000.

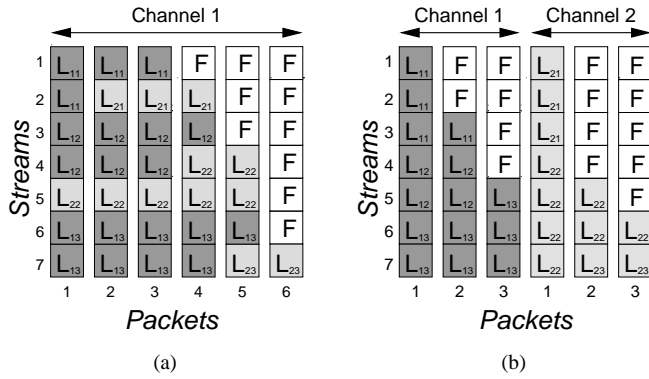


Fig. 2. Example packet arrangements, (a) CUEP, (b) IUEP. L_{ij} = data byte from layer j of object i .

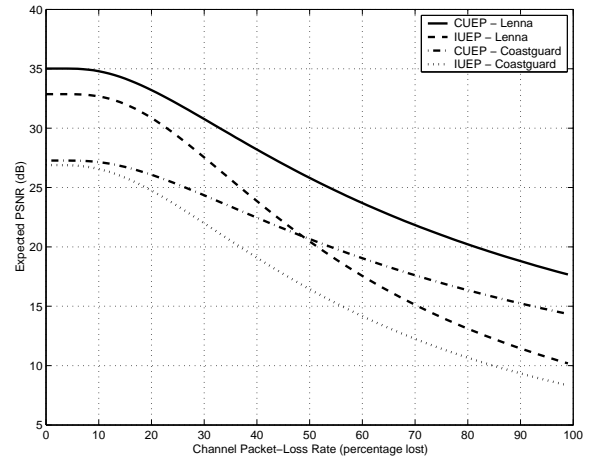


Fig. 5. Expected distortion vs. fraction of packets lost, protection schemes designed for 20% loss.



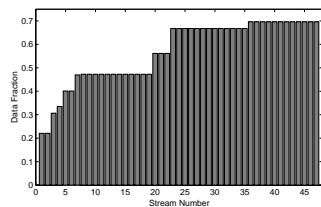
Fig. 3. Lenna and coastguard objects. Foreground = object 1, background = object 2.



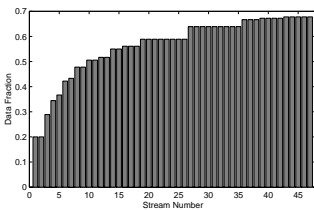
(a)



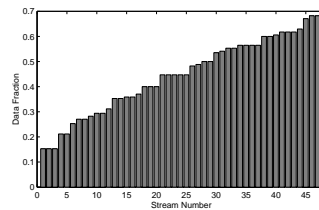
(b)



(a)



(b)



(c)

Fig. 4. FEC arrangement as fraction of stream devoted to data for lenna. (a) CUEP for interleaved bitstream, (b) IUEP for object 1, (c) IUEP for object 2.

Fig. 6. Reconstructed images at 50% packet loss. (a) CUEP, (b) IUEP.