EFFICIENT POST-COMPRESSION ERROR-RESILIENT 3D-SCALABLE VIDEO TRANSMISSION FOR PACKET ERASURE CHANNELS

†*Chih-Ming Fu, ‡Wen-Liang Hwang and †Chung-Lin Huang*

†Department of Electrical Engineering, National Tsing Hua University, Taiwan
‡Institute of Information Science, Academia Sinica, Taiwan

ABSTRACT

We propose an efficient error-resilient video transmission algorithm over packet erasure channels using optimal source and channel bit-allocation. This algorithm uses FEC and rate-distortion optimization to find the optimal allocation of source and channel bits in each quality layer. The packet loss probability is periodically reported to the video server. This method can also incorporate with any coding structure that generates a set of independent compressed bit-streams. The algorithm efficacy is demonstrated by simulations in which the video compression is an error-resilient 3D-SPIHT algorithm and the channel protection is provided by Reed-Solomon (RS) codes.

1. INTRODUCTION

In real-time transmission, both the encoder and decoder must be kept at low complexity, so they can adapt to varying environments. Current researches of multimedia transmission using joint source-channel coding keep the source and channel coders separate and combine each of them with some optimized parameters [1, 2, 3]. In this manuscript, the post-compression source-channel rate-distortion optimization algorithm is proposed to protect quality layers of an video form lossy channels. Our allocation of source and channel bits is applied after all the independent coding units have been compressed. We refer this method as the post-compression, source-channel rate-distortion (PCSC) optimization algorithm. The rate-distortion function of a compressed video is obtained by the compression process. We use the convex hull interpolation method to speed up our optimization process.

We show that the quality layer of this type of video compression can be easily and efficiently incorporated with the channel coding for error-resilience video transmission over a lossy environment. Our method can be easily incorporated in EBCOT-like coding structure. We denote "EBCOT-like" as an algorithm that uses an independent coding unit that generates an embedded bit-stream of a coding unit, and uses the post-compression rate-distortion (PCRD) optimization method to combine the contributions of code-blocks into quality layers [4]. We assume that a video is compressed by an EBCOT-like compression method, where pack-streams of quality layers are transmitted over a binary packet loss channel whose packet loss rate is periodically reported to the encoder.

The rest of this manuscript is organized as follows. In Section 2, we formulate our system . In Section 3, we present the proposed post-compression source and channel coding method. In Section 4, we demonstrate our simulation results. Finally, in Section 5, we present our conclusions.

2. SYSTEM FORMULATION

An EBCOT-like coding scheme contains two tiers. Tier-1 can be any embedded block-based coding technique that generates independent embedded bit-streams with truncation points. A tier-2 coder packs the bit-streams into pack-streams of quality layers. In an EBCOT-like coding structure, tier-2 only uses the source information from tier-1 and the coding constraints to generate quality layers. We propose that additional channel and packetization information be included in tier-2 for optimal source-channel coding. The two tier coding structure is shown in Figure 1. In tier-2, a rate-distortion optimization algorithm is applied to meet the coding constraints and optimize the source-channel bit-allocation to the quality layers of videos.

2.1. Error-Resilient and Scalable Video Coding

A video sequence is divided into groups of blocks, called video patches, and each patch is independently encoded. Consequently, an error will not propagate beyond a single patch, so error-resilience is improved [5]. The video quality scalability is imparted by the layering concept. The scalable bit-stream of each video patch is organized into quality layers. The number and the spacing of these layers determine the granularity of the video scalability. The sensitivity of the data to corruption clearly increases as we move from lower to higher quality layers. Therefore, we need to assign longer FEC codes to the higher quality layers.
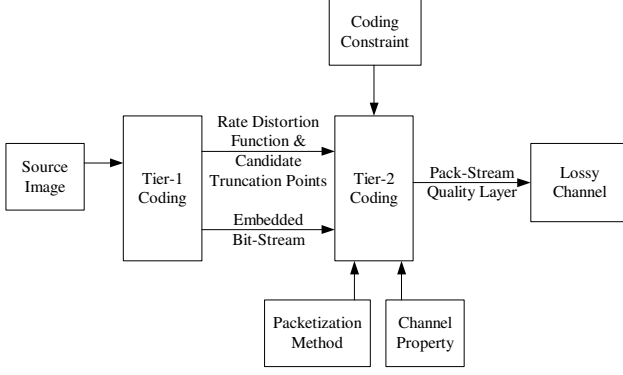
Fig. 1. Channel property and the packetization method are incorporated into tier-2 for post-compression joint source and channel coding.

2.2. Transmission Packetization

Because the packet is the basic transmission unit in the internet environment, packetization of the source and channel bits is one of the main factors that affects the performance of our post-compression source and channel bit-allocation. Figure 2 shows how an UEP for video patches [5] of different quality levels is generated, based on systematic RS codes. We assume that Q quality layers are generated and the video is divided into I patches. The number of source symbols contributed to the q-th layer from the i-th video patch is denoted as $l_{q,i}k_{q,i}$. These source symbols are arranged into $l_{q,i}$ different channel coding blocks, each of which has $k_{q,i}$ source symbols. As shown in Figure 2, symbols in the same quality layer, but belonging to different video patches, are interleaved into packets.

3. PCSC-BASED BIT-ALLOCATION

Our algorithm consists of a video compression, the error-resilient 3D set partitioning in hierarchical trees algorithm [5], and the channel protection is provided by Reed-Solomon (RS) codes. For all video patches in the q-th quality, we use the vectors $\bar{l}_q$ and $\bar{k}_q$ to respectively denote the numbers of channel coding blocks $\{l_{q,i}\}$ and source symbols $\{k_{q,i}\}$ of a channel coding block. Let $\bar{l}_q = [l_{q,1}, l_{q,2}, ..., l_{q,I}]^T$ and $\bar{k}_q = [k_{q,1}, k_{q,2}, ..., k_{q,I}]^T$. We also collect the above vectors from the first quality layer to the q-th quality layer: We have $\bar{L}_q = [\bar{l}_1, \bar{l}_2, ..., \bar{l}_q]$ and $\bar{K}_q = [\bar{k}_1, \bar{k}_2, ..., \bar{k}_q]$. Let each symbol contain b bits. The number of source bits assigned to the video patch i up to the quality layer q is

$$r_q^i = b \sum_{j=1}^{q} l_{j,i}k_{j,i}. \qquad (1)$$

We define the total bit constraint of each quality layer such that the number of source and channel bits allocated to video
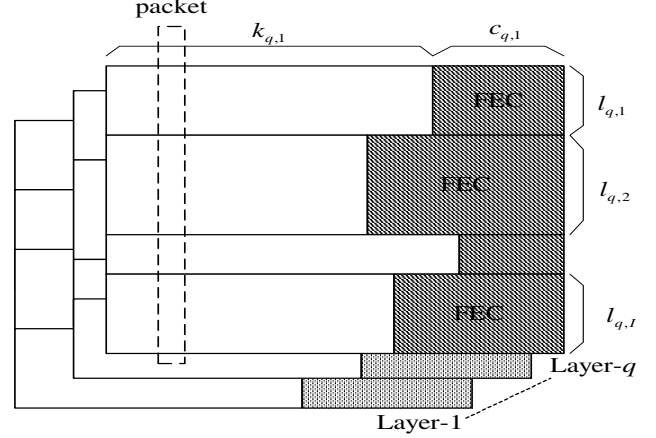
Fig. 2. Our packetizaion of source-channel bits at different video patches and quality layers. The number of the source and parity symbols of a block code at quality layer q and video patch i are $k_{q,i}$ and $c_{q,i}$, respectively. Data at different patches in the same quality layer is interleaved.

patches must meet the following constraint:

$$bn_0 \sum_{i=1}^{I} l_{q,i} \leq R_q, \qquad (2)$$

where R_q is the rate constrain of layer q; n_0 is the RS code length; and b is the symbol length (bit/symbol). b and n_0 are constants for all the quality layers.

If all the bits assigned to the q-th quality layer and the i-th video path are correctly decoded, the reduction of distortion is

$$\Delta D_{q,i}^s(r_{q-1}^i, r_q^i) = D_i^s(r_{q-1}^i) - D_i^s(r_q^i) \qquad (3)$$

where D_i^s is the distortion of video patch i; and r_q^i and r_{q-1}^i are defined in Equation 1. The expected distortion reduction of the q-th quality layer is

$$\overline{\Delta D}_q(\bar{L}_q, \bar{K}_q) = \sum_{i=1}^{I} \left[\prod_{j=1}^{q} P_c(n_0, k_{j,i}) \right] D_{q,i}^s(r_{q-1}^i, r_q^i). \quad (4)$$

The first component of the right term is related to the channel and the second is related to the source. The correct probability of an RS $(n_0, k_{j,i})$ decoder is characterized by $P_c(n_0, k_{j,i})$.

Equation 4 indicates that to decode the source bits of an video patch in quality layer q, we must recover all the block codes of the video patch from all previous quality layers. This equation also measures the expected distortion reduction for one quality layer. The cumulative distortion reduction for all quality layers is

$$\overline{\Delta D}(\bar{L}_Q, \bar{K}_Q) = \sum_{q=1}^{Q} \overline{\Delta D}_q(\bar{l}_q, \bar{k}_q). \qquad (5)$$

We can formulate the source-channel optimization problem as a search for $\bar{L}_Q$ and $\bar{K}_Q$ that maximizes the expected quality in Equation 5 under the rate constraint at each quality layer, given in Equation 2. The sub-optimal solution of the above approximation can be found by finding the source and channel symbol assignment at the first quality layer, and then using the assignment to find the second quality layer, etc. With this approach, we need to find the optimal channel and source symbol assignment of a quality layer.

Finding the source and channel assignment of a quality layer is an optimization problem whose formulation is expressed as

$$
\max_{\bar{l}_q, \bar{k}_q} \overline{\Delta D}_q([\bar{l}_1^*, \cdots, \bar{l}_{q-1}^*, \bar{l}_q], [\bar{k}_1^*, \cdots, \bar{k}_{q-1}^*, \bar{k}_q])
$$

$$
\text{subject to } bn_0 \sum_{i=1}^{I} l_{q,i} \leq R_q, \qquad (6)
$$

where $\overline{\Delta D}_q$ is defined in Equation 4. When solving the source-channel assignment for the q-th quality layer, the bit assignments of the previous layers are obtained and represented as $\bar{l}_j^*$ and $\bar{k}_j^*$ with $j = 1, \cdots, q-1$. A solution can be found by taking the derivative of the following rate-distortion constraint with respect to $l_{q,i}$, $k_{q,i}$ along with the Lagrange multiplier, and setting the results to zero.

$$
\Lambda_q(\bar{l}_q^*, \bar{k}_q^*) = \max_{\bar{l}_q, \bar{k}_q} \left\{ \overline{\Delta D}_q(\bar{L}_q^*, \bar{K}_q^*) + \lambda(R_q - \sum_{i=1}^{I} bn_0 l_{q,i}) \right\}.
$$

Note that although $l_{q,i}$ and $k_{q,i}$ are integers, we use the derivative to take advantage of the simplicity of real number calculus. We thus have

$$
\frac{\partial \Lambda_q(\bar{l}_q, \bar{k}_q)}{\partial l_{q,i}} = 0, \frac{\partial \Lambda_q(\bar{l}_q, \bar{k}_q)}{\partial k_{q,i}} = 0, \qquad (7)
$$

After some derivations, we show the results as follows:

$$
\prod_{j=1}^{q} P_c(n_0, k_{j,i}) bk_{q,i}(D_i^s)'(bl_{q,i}k_{q,i} + r_{q-1}^{i*}) + \lambda bn_0 = 0,
$$

$$
\qquad (8)
$$

$$
\frac{\partial P_c(n_0, k_{q,i}) \Delta D_{q,i}^s(r_{q-1}^i, r_q^i)}{\partial k_{q,i}} = 0. \qquad (9)
$$

We use the bisection method to search for the optimal λ^* within the interval $[\lambda_{min}, \lambda_{max}]$ and to fit the rate constraint $R_q - bn_0 \sum_{l=1}^{I} l_{q,l} = 0$.

The efficacy of this approach is based on the convex rate-distortion function for all video patches. For a specific source compression algorithm, the corresponding rate-distortion curve is not usually a smooth convex function. Thus, we pre-process a rate-distortion curve by finding the convex hull of the curve, and use the resultant curve for the proposed algorithm.

The proposed algorithm that finds the optimal solution of Equations 8 and 9 is accurate and efficient. We show that for a given λ the expected PSNR, as a function of the channel coding rate and the bits per pixel for a typical video patch, is concave. Our algorithm alternately applies the following two steps until we reach a point that has the maximal value. The first step moves along a curve of a fixed channel coding rate and stops at the point that satisfies Equation 8, while the second step moves along a curve of fixed bits per pixel searching for a coding rate that satisfies Equation 9.

4. SIMULATION RESULTS

In our simulations, we divided an video into several patches [5]. The bit-stream of each patch is independently produced by applying the SPIHT algorithm without using an arithmetic coder. The common parameters for our simulations are: The original forman.yuv video size is 352 by 288 pixels with 16 frames, and each video patch size is $88 \times 72 \times 16$. We used 3 levels of spatial wavelet decomposition with 9-7 biorthogonal filters and 3 levels of temporal wavelet decomposition with haar filter. In order to compensate for non-energy preservation of the biorothogonal filters, we use a weighting factor, to calculate MSE in wavelet domain, and implement the RS codes over GF(16). This means that each symbol is a nibble, having b = 4 bits. For example, the $(15, 9)$ code produces codewords with 15 nibbles (60 bits) each, and is able to locate and correct errors in any 6 nibbles.

Figure 3 shows the expected PSNR (EPSNR) of various packet loss protection methods as a function of bit-rates in an environment which has a 5% packet loss probability. The expected PSNR is defined as

$$
EPSNR(Q) = 10 \log_{10} \frac{255^2}{D^s(0) - \overline{\Delta D}(\bar{L}_Q, \bar{K}_Q)},
$$

where $\overline{\Delta D}(\bar{L}_Q, \bar{K}_Q)$ is the cumulative distortion reduction defined in Equation 5 and $D^s(0)$ is the variance of the whole vidoe.

The side information $[\bar{L}_q, \bar{k}_q]$ corresponds to a 0.002 bpp. As the side information of our source-channel coding structure is small and could be reduced even further by using compression algorithm. We can ignore it in the plots of the simulation results. As the number of quality layers increases, the side information increases. However, at layers of higher bit-rates, the channel coding rate approaches the theoretical results,and becomes a constant for all video patches at these layers (thus, we need not to send k in these layers). Also at a higher bit-rate, the rate-distortion curves becomes flat, and the source bit allocation becomes constant, as does the parameter l.

Figure 3 shows the performance of different methods. Curve-1 illustrates a lossless environment without channel
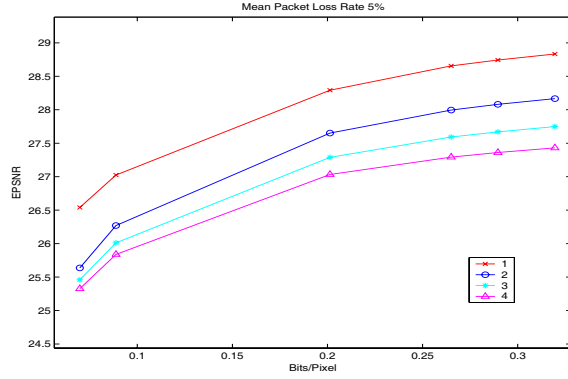
Fig. 3. Comparison of the performance of various methods with 6 quality layers.
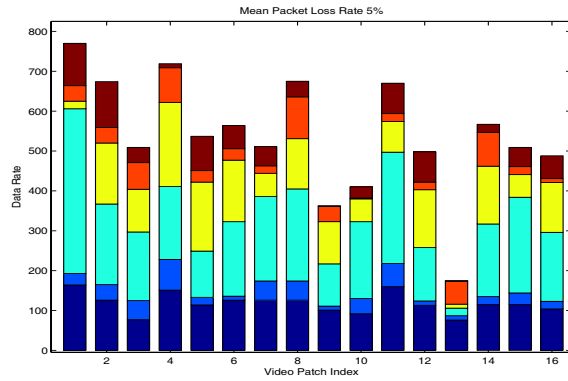

Fig. 4. The distribution of the optimal bit-rate assignment to video patches and quality layers of the foreman video.

coding. Curve-2 shows the performance of our method. Curves-3 and Curves-4 have the same channel coding rate ($k = 11$) at all quality layers and different source bit assignments. Curve-4 shows the equal source bits assignment to all video patches, while Curve-3 demonstrates the optimal source bits allocation. In a lossy environment without any channel coding, the EPSNR is about 7dB. Figure 4 illustrates the optimal source-channel bits allocation for each video patch at different quality layers. The optimal channel coding rates of each video patch at different quality layers are shown in Figure 5. It illustrates that more channel bits (lower code rate) are used in lower quality layers.

5. CONCLUSION

We propose an effective algorithm to produce source and channel optimized pack-streams of quality layers over packet erasure channels. Our algorithm can be incorporated to any EBCOT-like coding structure in which tier-1 yields a set of dependently encoded embedded bit-streams with trunca-
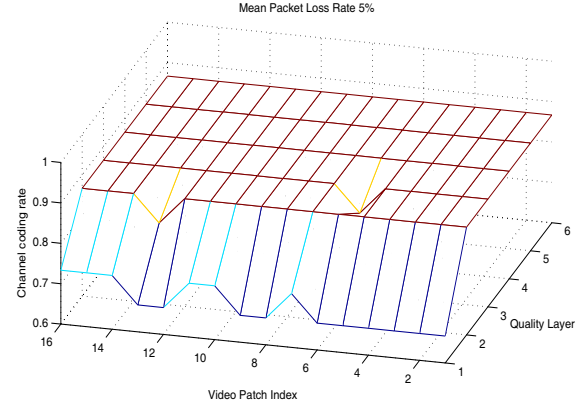
Fig. 5. The distribution of optimal channel coding rate of a foreman video. The video patches are numbered and arranged as a line.

tion points. The truncation points are sampled on a rate-distortion curve at certain bit-rates. We demonstrate the efficacy of our method in using error-resilient SPIHT and RS codes. Assigning optimal rates to quality layers is worth further study.

6. REFERENCES

[1] V. M. Stankovic, R. Hamzaoui, and Z. Xiong, "Efficient channel code rate selection algorithms for forward error correction of packetized multimedia bitstreams in varying channels," *IEEE Transcation on multimedia*, vol. 6, pp. 240–248, April 2004.

[2] J. K. Russell, M. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the internet using unequal error protection," *IEEE Transations on Image Processing*, vol. 12, pp. 121–131, Feb. 2003.

[3] N. V. Boulgouris, N. Thomos, and M. G. Strintzis, "Transmission of images over noisy channels using error-resilient wavelet coding and forward error correction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 1170–1181, Dec. 2003.

[4] D. Taubman, "High performance scalable image compression with ebcot," *IEEE Transactions on Image Processing*, vol. 9, pp. 1158–1170, July 2000.

[5] S. Cho and W. A. Pearlman, "A full-featured, error-resilient, scalable wavelet video codec based on the set partitioning in hierarchical trees (spiht) algorithm," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 157–171, March 2002.