OPTIMAL PACKETIZATION OF FINE GRANULARITY SCALABILITY CODESTREAMS FOR ERROR-PRONE CHANNELS

Bin B. Zhu[1], Yang Yang[2], Chang Wen Chen[3], Shipeng Li[1]

[1]Microsoft Research Asia, Beijing 100080, China
[2]Dept. of Elec. Eng. & Info Sci., Univ. of Sci. & Technol. of China, Hefei, Anhui 230027, China
[3]Dept. of Elec. and Computer Eng., Florida Inst. of Technology, Melbourne, Florida 32901, USA
[1]{binzhu, spli}@microsoft.com, [2]wdscxsj@ustc.edu, [3]cchen@fit.edu

ABSTRACT

An optimal source-channel packetization scheme for MPEG-4 Fine Granularity Scalability (FGS) codestreams is proposed in this paper. The channel is modeled with a uniform error distribution to the enhancement layer transmission. A cost function that models the error expansion for a MPEG-4 FGS stream is derived, and then used in the optimal packetization problem subject to the same overhead as the conventional packetization scheme. An efficient scheme to find the optimal solution is described, which takes time similar to encoding an MPEG-4 FGS codestream. Experiments show that our scheme has up to 1.96 dB gain over the conventional packetization scheme.

1. INTRODUCTION

Scalable coding with Fine Granularity Scalability (FGS) such as MPEG-4 FGS [1][2] or its variant Progressive FGS [3] has been designed for multimedia streaming applications over heterogeneous networks with varying bandwidths. MPEG-4 FGS encodes a video sequence into two layers: a base layer and an enhancement layer. The base layer is encoded with a non-scalable coder to provide the lowest quality and bitrate for a scalable codestream. The enhancement layer is encoded in a scalable manner to provide enhancement to the base layer. More precisely, the difference between the original frame and the reconstructed frame from the base layer is encoded bitplane-wise from the Most Significant Bitplane (MSB) to the Least Significant Bitplane (LSB). Each bitplane of an 8-by-8 block's DCT coefficients is zigzag-ordered, converted to (RUN, EOP) symbols, and coded with variable-length coding to produce an enhancement layer codestream, where RUN is the number of consecutive zeros before a nonzero value and EOP indicates if there is any non-zero value left on the current bitplane for the block. The enhancement layer can be arbitrarily truncated to adapt to varying bandwidths.

Networks are not perfect. When multimedia is transmitted over a network, errors are introduced during transmission, resulting in lowered quality of received signals. Degradation can be severe with some error-prone such as wireless networks. Many schemes have been developed to deal with transmission errors and packet losses in streaming applications for MPEG-4 FGS and other scalable codestreams. A simple scheme used in MPEG-4 [1][2] is to packetize compressed video data into Video Packets (VPs) to prevent extensive error expansion. Necessary decoding information is inserted into each VP for correct decoding of the VP even if some preceding VPs are lost or corrupted with errors. The VP packetization strategy adopted by MPEG-4 is a uniform, periodic scheme which inserts periodic resynchronization markers "uniformly" throughout the bitstream. In other words, the length of a VP is based on the number of bits contained in that VP. If the number of bits contained in the current VP exceeds a predetermined threshold, then a new VP is created at the start of the next macroblock (MB). For the MPEG-4 FGS enhancement layer, the bitplane boundary delimiter also serves as a resynchronization marker. This conventional packetization scheme introduces significant dependency among VPs. When one VP is lost or corrupted, other VPs can still be affected due to the inter-VP dependency. Cai et al. [4] proposed two improved packetization schemes designed to minimize inter-VP dependency. One scheme is a mixture of horizontal and vertical packetization that multiple bitplanes from multiple blocks can contribute to a VP with a careful boundary alignment. The other is a vertical packetization which removes inter-VP dependency completely. A codestream generated by these two schemes is not compliant to the MPEG-4 FGS standard. A compliant decoder may not be able to correctly parse the codestream. Many more complex and advanced technologies combined with channel coding such as unequal error protection [5][6] have also been proposed to address the problem. Optimal packetization was studied in [7] for embedded bitstreams generated by a wavelet-based coding technology, and in [8] for general scalable bitstreams to transmit over erasure channels. They have taken into consideration most important issues such as packet dependency, time constraint, and error protection in optimal streaming of scalable bitstreams.

In this paper, we consider an optimal source-channel packetization problem for MPEG-4 FGS bitstreams: Given the characteristics of the channel that an MPEG-4 FGS bitstream is to be transmitted over as well as allowed packetization overhead, what is the optimal packetization that is still MPEG-4 FGS compliant? The channel is modeled as an erroneous channel with a uniform error probability for transmission of the MPEG-4 FGS enhancement layer. The base layer is assumed to be heavily protected that is virtually error-free during transmission. We exploit the intrinsic dependency inside an MPEG-4 FGS bitstream to come up with a simple cost model. Then we give an efficient scheme to find the optimal solution to the cost function. Experiments show that our optimal packetization has a gain of up to 1.96 dB over the conventional packetization scheme adopted by MPEG-4 with the same packetization overhead in transmission over the assumed channel.

The rest of this paper is organized as follows. In Section 2 we introduce the error expansion modeling and the cost function. An efficient scheme to find the optimal packetization is described in Section 3, which is followed by experimental results in Section 4 to compare with the MPEG-4 FGS packetization method. We conclude in Section 5 with summary and future work.

2. COST FUNCTION FOR MPEG-4 FGS

2.1. Error Expansion

As we have mentioned in the previous section, MPEG-4 FGS converts each bitplane of a block to (RUN, EOP) symbols, and then encodes with Variable Length Coding (VLC). The sign bit of a coefficient is inserted into the bitstream right after the VLC coding of the MSB of the coefficient. The output bitstreams from all the blocks in a frame at a given bitplane are concatenated according to the block index into a VP until being terminated by either the end of the bitplane or a resynchronization marker. Bitplane delimiter *fgs_bp_start_code* is placed at fixed positions: the start of a bitplane, while the resynchronization marker *fgs_resync_marker* can be placed anywhere aligned with MBs. The index of the first MB in a VP is inserted right after the marker *fgs_resync_marker*. For a given number of bitplanes to be coded, and a given number of VPs, the overhead of packetization is fixed.

When there is an error in a VP header, the whole VP is likely to be decoded incorrectly and is dropped by the decoder. The bitstream is searched to locate the delimiter of the next VP. If there is an error in a VP data stream, the bitplane of a block that the error occurs is corrupted, and is likely to result in a wrong number of decoded bits for the block's bitplane. The bitplanes of subsequent blocks are therefore likely to get corrupted too. The lower bitplanes of these blocks may not be correctly decoded since if a sign bit is lost due to error expansion in the given bitplane, a lower bitplane with non-zero bit corresponding to the coefficient of the lost sign bit may incorrectly treat the next bit as the sign bit for the coefficient, resulting in a misaligned bitstream and wrong decoded data.

The aforementioned error expansion characteristics of MPEG-4 FGS are approximated with the following modeling which is used in deriving the cost function for our optimal packetization: when an error occurs in a block's bitplane bitstream, the bitplanes of the current and the subsequent blocks in the same VP are dropped in final reconstruction, while the bitplanes of preceding blocks in the VP are not affected. Within a block, if a bitplane contains an error, all the lower bitplanes are also dropped in reconstruction for the block. For simplicity, such an error expansion within a block is taken into consideration in the distortion of a block's bitplane but ignored in calculating the probability that a bitplane of a block contributes in reconstruction. The effect that an error occurs in VP headers is ignored in our modeling.

2.2. Cost Function

Let the *k*-th VP in the *j*-th bitplane be denoted by VP_k^j which has M_k^j blocks. Assume that the *j*-th bitplane has N^j VPs, the total number of VPs in a frame is N, and the number of bitplanes is L. Let the *i*-th Contribution of a Block's Bitplane (CBB) in VP_k^j be $CBB_{i,k}^j$, which has $l_{i,k}^j$ bits. Let $D_{i,k}^j$ be the distortion reduction in the reconstructed frame due to $CBB_{i,k}^j$ and the lower bitplanes of the same block. Assume that the channel causes a uniform error distribution over all the bits in a bitstream of the enhancement layer, and the Bit Error Probability (BEP) is p_e. Also assume that the base layer is error-free in transmission, which corresponds to a practical streaming application under bandwidth constraint that the base layer is heavily protected while the enhancement layer is unprotected (or equally protected by lower communication layers). Unlike other approaches which assume that lower communication layers drop a whole transport packet (thus a lost packet) once an error is detected in the packet, we assume that lower communication layers pass every received packet up to the application layer and let the application make a decision on how to deal with an error packet. This is very desirable in streaming scalable multimedia since most data in an error packet may be still usable by a scalable decoder. This is a very unique feature of an FGS scalable bitstream. A consequence is that a scalable decoder must be strong in error detection and robust to errors.

Since we have ignored the inter-VP dependency in calculating the probability $P_{i,k}^j$ that $CBB_{i,k}^j$ is corrupted by an error in $CBB_{i,k}^{j}$, $P_{i,k}^{j}$ depends only on the preceding number of bits in VP_{k}^{j}, and is given by:

$$P_{i,k}^{j} = p_r^{\sum_{c=0}^{i-1} l_{c,k}^{j}} (1 - p_r^{l_{i,k}^{j}}),$$

where $p_r = 1 - p_e$. The average distortion for the frame is

$$\begin{aligned} J &= \sum_{j=0}^{L-1} \sum_{k=0}^{N^j-1} \sum_{i=0}^{M_k^j-1} P_{i,k}^{j} \cdot (\sum_{c=i}^{M_k^j} D_{c,k}^{j}) \\ &= \sum_{j=0}^{L-1} \sum_{k=0}^{N^j-1} \sum_{i=0}^{M_k^j-1} p_r^{\sum_{c=0}^{i-1} l_{c,k}^{j}} (1 - p_r^{l_{i,k}^{j}}) \cdot (\sum_{c=i}^{M_k^j} D_{c,k}^{j}) \quad \text{Eq. (1)} \\ &= \sum_{j=0}^{L-1} \sum_{k=0}^{N^j-1} \sum_{i=0}^{M_k^j-1} (1 - p_r^{\sum_{c=0}^{i} l_{c,k}^{j}}) \cdot D_{i,k}^{j}, \end{aligned}$$

which is the cost function we want to minimize for the frame. Although the cost function for multiple frames can also be derived, we focus on finding optimal packetization for each individual frame in this paper. We want the packetization overhead remains constant in fining optimal packetization. This condition is virtually equivalent to assuming that the total number of VPs in the frame is a constant. Therefore our problem is to find the optimal values $\hat{N}^j$ and $\hat{M}_k^j$ such that

$$(\hat{N}^j, \hat{M}_k^j) = \arg\min(J) \qquad \text{Eq. (2)}$$

subject to the condition:

$$N = \sum_{j=0}^{L-1} N^j = const. \qquad \text{Eq. (3)}$$

This is equivalent to the problem that each resynchronization marker *fgs_resync_marker* in the codestream generated by the conventional packetization method of MPEG-4 FGS is moved around among all legal positions such that the reconstructed frame from the received data over a network of a uniform error distribution has the highest PSNR value on the average.

By definition, the distortion reduction $D_{i,k}^{j}$ is given by

$$D_{i,k}^{j} = \sum_{c=0}^{j} n_c 2^{2(L_{\max}-1-c)} \qquad \text{Eq. (4)}$$

within a constant factor, where n_c is the number of non-zero bits for the c-th bitplane of the block, and $L_{\max}$ is the maximum number of bit levels, and *0*-th bitplane corresponds to MSB.

3. OPTIMAL PACKETIZATION

The optimization is a discrete problem. All the data except the optimal values to be found are available from the actual bitstream of the frame. There are a finite but large number of possible combinations. Therefore there always exists a solution theoretically. When the set of combinations is large, it may be too complex to enumerate all possible combinations to find out the optimal solution. An efficient algorithm is needed.

Let us first look at the problem for a simple case that there is only one bitplane: $L = 1$. The number of VPs for the bitplane is then $N^0 = N = const$. The following proposition lays the foundation that our efficient scheme will always find the optimal packetization:

Proposition 1: *Assume* $L = 1$. *If a set* $\Gamma_N = \{\hat{M}_0^0, \cdots, \hat{M}_{N-2}^0, \hat{M}_{N-1}^0\}$ *is an optimal solution to Eq. (2) subject to* N *VPs, then the first* $N-1$ *partition points still optimally partition the remaining bitstream with the last substream delimited by* $\hat{M}_{N-2}^0$ *and* $\hat{M}_{N-1}^0$ *(i.e., the last optimal VP for the problem with* N *VPs) being truncated, subject to* $N-1$ *VPs.*

This proposition is easy to prove, since otherwise the optimal partition to the problem with $N-1$ VPs combined with the truncated VP has smaller cost than the optimal partition to the problem with N VPs, contradicting the assumption. The following algorithm finds out the optimal partition with N VPs for the case $L = 1$.

Algorithm 1 (single bitplane case): the minimal cost is returned.

op_1 ***(N,*** **NB*)*** // Inputs: *N*, *NB* (# of MBs). $N \le NB$ **.**

```
{    if (N == 1) return the cost as a single VP;
     minCost = positive infinite;
     for (k = N-1; k < NB; ++k) {
          t = op_1(N-1, k) + cost of the rest (NB – k) MBs
                         as a single VP;
          If (t < minCost) {
               minCost = t; Remember the current partition.
          }
     }
     return minCost;
}
```

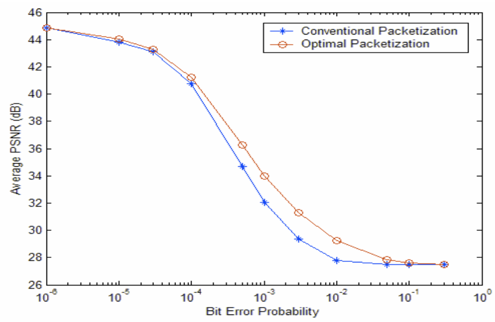

Figure 1: PSNR at different bit error probabilities.

Algorithm 1 takes exponential time. Dynamic programming [9] is used to bring it down to $O(N * NB)$ time. The next algorithm finds the optimal packetization for the multiple bitplane case by increasing one VP each round starting from L VPs (note that each bitplane contains at least one VP delimited by *fgs_bp_start_code*).

The additional VP in each round is assigned to the bitplane which gives the maximum cost reduction.

Algorithm 2 (general case):

op_gc (N, L) // Inputs: *N, L. N ≥ L*

```
{   for (n = L+1; n <= N ; ++n) { // n VPs
        maxRedu = 0;
        maxbp = -1;
        for (bp = 0; bp < L-1; ++bp) {
            t = cost reduction calculated by op_1() if the
                    number of VPs in the bp-th bitplane is
                    increased by 1.
            if (t>maxRedu): maxRedu = t; maxbp = bp;
        }
        Add one VP to the bitplane of index maxbp;
    }
}
```

4. EXPERIMENTS

We have implemented the proposed scheme and applied it to MPEG-4 FGS bitstreams generated from MPEG's standard QCIF sequences. The base layer was encoded to about 100 kbps with the original frame rate, while the enhancement layer was encoded at about 2.1 Mbps. The nominal packet size was set to 500 bits for the conventional packetization scheme. The number of VPs for each frame generated by the conventional scheme is input to our scheme to produce the optimal packetization constrained with the same packetization overhead as the conventional scheme. In each round, a random seed was used to generate random error locations with a uniform distribution. The bits at the error locations in the two bitstreams generated with both packetization schemes were flipped, and then decoded.

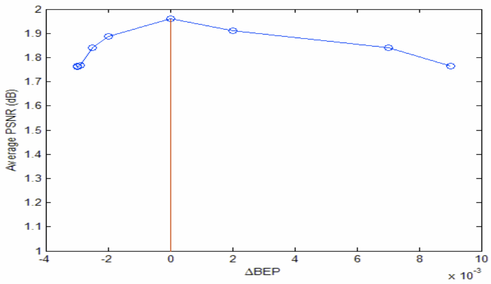

Figure 2: PSNR gain for mismatched BEPs: the network's BEP is 0.003 while the packetization model's BEP ranges from 0.000001 to 0.012.

Figure 1 shows the average PSNR of the two schemes for the sequence Foreman, with the Bit Error Probability (BEP) in the range from 0.000001 to 0.3. At the low BEP end, errors are rare so that the PSNR is about the same as the case without any error. The PSNR difference is about 0. At the high BEP end, there are so many errors that almost all data in the enhancement layer were dropped, resulting in almost no difference. In the middle range of BEP, our scheme shows a significant gain in PSNR over the conventional scheme, with the maximum gain of 1.96 dB at BEP = 0.003 for the sequence.

In real applications, it may be difficult to have a good estimation of a channel's BEP. Another experiment was conducted to evaluate the performance when the model mismatches the actual network's BEP. Figure 2 shows the PSNR gain of our scheme over the conventional scheme when the network's BEP is at 0.003 while the model's BEP ranges from 0.000001 to 0.012. Our scheme still shows more than 1.7 dB gain. In comparison, the gains of the Cai et al.'s packetization schemes [4] over the conventional packetization scheme are up to 1.5 dB.

Preliminary results show that time spent on finding the optimal packetization is close to the time spent on encoding a frame at the given experimental setting.

5. CONCLUSION

We have described a simple model to approximate transmission networks and the error expansion in a MPEG-4 FGS bitstream, and a corresponding cost function. An efficient scheme has been proposed which can always find the optimal packetization in minimizing the cost function subject to the same overhead as the conventional packetization scheme. Experiments showed that our scheme could gain up to 1.96 dB over the conventional scheme. Future work includes unequal error protection and other error distribution channels.

REFERENCES

[1] *Coding of Audio-Visual Objects, Part-2 Visual, Amendment 4: Streaming Video Profile*, ISO/IEC 14496-2/FPDAM4, July 2000.

[2] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. on Circuits and Systems for Video Technol.,* vol. 11, no. 3, pp. 301–317, 2001.

[3] F. Wu, H. Sun, G. Shen, S. Li, Y.-Q. Zhang, B. Lin, and M.-C. Lee, "SMART: An Efficient, Scalable, and Robust Streaming Video System," *EURASIP J. Appl. Signal Proc.* 2004:2, pp. 192–206, 2004.

[4] H. Cai, G. Shen, Z. Xiong, S. Li, and B. Zeng, "An Optimal Packetization Scheme for Fine Granularity Scalable Bitstream," *IEEE Intl. Symposium Circuits and Systems*, vol. 5, pp. 641-644, May 2002.

[5] H. Cai, B. Zeng, G. Shen, and S. Li, "Error-Resilient Unequal Protection of Fine Granularity Scalable Video Bitstreams," *IEEE Intl. Conf. on Comm.*, vol. 3, pp. 1303-1307, 2004.

[6] S. Dumitrescu, X. Wu, and Z. Wang, "Globally Optimal Uneven Error-Protected Packetization of Scalable Code Streams," *IEEE. Trans. Multimedia*, vol. 6, no. 2, pp. 230-239, April, 2004.

[7] X. Wu, S. Cheng, and Z. Xiong, "On Packetization of Embedded Multimedia Bitstreams," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 132-140, March 2001.

[8] P. A. Chou and Z. Miao, "Rate-Distortion Optimized Streaming of Packetized Media," *Microsoft Technical Report*, MSR-TR-2001-35, Feb. 2001.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press.