OPTIMAL ERROR PROTECTION OF PROGRESSIVELY COMPRESSED 3D MESHES

Shakeel Ahmad and Raouf Hamzaoui

Department of Computer and Information Science, University of Konstanz, Germany

ABSTRACT

Given a number of available layers of source data and a transmission bit budget, we propose an algorithm that determines how many layers should be sent and how many protection bits should be allocated to each transmitted layer such that the expected distortion at the receiver is minimum. The algorithm is used for robust transmission of progressively compressed 3D models over a packet erasure channel. In contrast to the previous approach, which uses exhaustive search, the time complexity of our algorithm is linear in the transmission bit budget.

1. INTRODUCTION

The robust transmission of multimedia data over unreliable networks can be achieved with forward error correction, ARQ protocols, error resilient encoding, error concealment techniques, or any combination thereof. While many systems were proposed for images and video (see the surveys [1, 2]), only a few [3, 4, 5] were dedicated to 3D graphic models.

In this paper, we focus on the system of [5], which is the only one based on forward error correction. This system starts by compressing the 3D model with a progressive compression scheme to one base layer and a number of update layers. This can be done, for example, with the *Compressed Progressive Meshes* (CPM) coder [6]. Then each layer is protected by applying forward error correction and sent over the channel in a block of packets (BOP) consisting of information symbols and protection symbols. Given a total transmission bit budget, the performance of the system is optimized by finding the number of layers that should be transmitted and the number of protection symbols allocated to each transmitted layer such that the expected distortion at the receiver is minimum. To solve this optimization problem, Al-Regib *et al.* [5] use exhaustive search whose complexity is reduced by constraining the number of protection symbols assigned to a given layer to be greater than or equal to the number of protection symbols assigned to the next layer. We show that this monotonicity constraint on the number of protection symbols can lead to poor performance and propose a dynamic programming algorithm that solves the unconstrained optimization problem in linear time and space. Our algorithm is inspired from the algorithm of [7]. However, there are important differences between the two algorithms. First, the algorithm of [7] was developed for embedded source bitstreams, which can be decoded regardless of the number of received symbols. In contrast, the source coder in the system of [5] allows the decoding only at a given number of truncation points, each of which corresponds to a layer. Second, the packet size is fixed a priori in [7], while it can vary with the BOP in [5]. Third, the number of information symbols allocated to a BOP is variable in the system of [7], whereas it is fixed in the system of [5]. Fourth, the number of transmitted BOPs is fixed in [7], whereas it must be optimized in [5]. Finally, our algorithm has linear time and space complexity, while the algorithm of [7] has quadratic time and space complexity.

2. PREVIOUS WORK

Consider a 3D model progressively compressed to M layers consisting of one base layer and $M - 1$ update layers. To decode a given layer, all information symbols in this layer and in the previous layers must be available. Let s_j denote the number of symbols (for example, bytes) in the jth layer ($j = 1, \dots, M$). For $j = 1, \dots, L$ ($1 \leq L \leq M$), the jth layer is packetized into a group of k_j horizontal information packets of size m_j symbols each. Then the same systematic Reed-Solomon code (or shortened Reed-Solomon code) of length N is applied vertically on each of the m_j groups of k_j information symbols, yielding a BOP consisting of N channel packets of size m_j symbols each. The number N of channel packets in a BOP is fixed, while both k_j and m_j are variable and satisfy $1 \leq k_j \leq N$, $k_j m_j = s_j$, and $(N - k_j)m_j = c_j$, where c_j is the number of protection symbols assigned to the jth layer (Table 1).

i	i	i	i	i	i	i
i	i	i	i	i	i	i
x	x	x	x	i	i	x
x	x	x	x	x	x	x
x	x	x	x	x	x	x

Table 1. $L = 3$ layers are sent in three BOPs of $N = 5$ packets each. The length of the packets is $m_1 = 4$, $m_2 = 2$, and $m_3 = 1$ in the first, second, and third BOP, respectively. Information symbols are denoted by i and protection symbols by x.

Because Reed-Solomon codes are maximum distance separable codes, when layer j is transmitted over a packet erasure channel, the receiver can recover all s_j information symbols if no more than $N - k_j$ channel packets are lost out of the N transmitted ones.

Suppose now that the total transmission budget is fixed to $S = s_1 + \cdots + s_M$. Then one can either send all lay-

1-4244-0367-7/06/$20.00 ©2006 IEEE

ers without protection, or the first $M-1$ layers with a total protection budget $c_1+\cdots+c_{M-1}=s_M$, or the first $M-2$ layers with a total protection budget $c_1+\cdots+c_{M-2}=s_M+s_{M-1}$, etc. That is, the total transmission budget is kept constant by trading off information symbols for protection symbols. When L BOPs are sent ($1\le L\le M$) and an *error protection allocation* $\mathbf{c}_L=(c_1,\ldots,c_L)$ is used, the expected distortion is

$$E(\mathbf{c}_L)=(1-B_1(c_1))E_0+\sum_{j=2}^{L}E_{j-1}(1-B_j(c_j)) \times\prod_{k=1}^{j-1}B_k(c_k)+E_L\prod_{k=1}^{L}B_k(c_k) \quad (1)$$

where $B_j(c)$ is the probability of successfully recovering all the information symbols in the jth layer when $c_j=c$, and E_j is the distortion if the first j layers are reconstructed. Here we assume that packet losses are independent in different BOPs. Note that E_j is the error between the reconstructed 3D model if all M layers are decoded and the reconstructed 3D model if only the first j layers are decoded. The goal is to determine the number of transmitted layers L and the associated error protection allocation $\mathbf{c}_L=(c_1,\ldots,c_L)$ such that the expected distortion (1) is minimum. Al-Regib *et al.* [5] compute a solution to this problem by full search. This is done as follows. For all values of $L,\ L=M,\ldots,1$, with corresponding protection budget $C_L,\ C_L=0,s_M,s_M+s_{M-1},\ldots,s_M+s_{M-1}+\cdots+s_2$, all possible allocations $\mathbf{c}_L=(c_1,\ldots,c_L)$ such that $c_1+\cdots+c_L=C_L$ are tested, and the best one is selected. To reduce the complexity of this exhaustive search, they propose to allow only solutions that satisfy the constraint $c_1\ge\ldots\ge c_L$. Moreover, they suggest to stop the iteration on L in the algorithm as soon as the expected distortion of the best allocation for L is greater than that for the previous L.

The approach of [5] has a major drawback. When C_L is large, the number of possible L-tuple candidates $(c_1,\ldots,c_L)$ that satisfy the constraint $c_1+\cdots+c_L=C_L$ and $c_1\ge\ldots\ge c_L$ is too large to allow an exhaustive search. For example, when $L=11$, the number of candidates is 42, 560, 20298, 2012069, and 470259534 for $C_L=10,20,40,80$, and 160, respectively. Because of the BOP packetization constraints $1\le k_j\le N, k_jm_j=s_j$, and $(N-k_j)m_j=c_j$, not all candidates are admissible. But this is not an advantage as usually only a few candidates will be admissible, leading to a poor performance of the system. Worse, in many cases, there will be no admissible allocations. For example, when $s_1=4269$, $N=100$, and $C_L=2650$, it is not possible to build a BOP for the first layer.

3. PROPOSED SOLUTION

In this section, we first present a flexible BOP packetization technique. Then we provide a linear-time algorithm that finds an optimal number of layers to be transmitted and a corresponding optimal error protection allocation. In contrast to [5], we do not assume that the number of protection symbols is nonincreasing. Moreover, we do not assume that the transmission budget is fixed to $s_1+\cdots+s_M$. Instead, we compute our solution for any arbitrary total transmission budget.

i	i	i	i	i	i	i
i	i	i	i	i	i	i
i	i	i	i	i	i	i
i	i	i	i	i	i	i
i	i	i	i	i	i	d

i	i	i	i	i	i	i	i	i
i	i	i	i	i	i	i	i	i
i	i	i	i	i	i	i	i	d
i	i	i	i	i	i	i	i	d
x	x	x	x	x	x	x	x	x

i	i	i	i	i	i	i	i	i	i	i	i
i	i	i	i	i	i	i	i	i	i	i	d
i	i	i	i	i	i	i	i	i	i	i	d
x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x

Table 2. Three of the five possible packetizations of layer j into a BOP when $N=5$ and $s_j=34$. The letter d denotes a dummy symbol.

3.1. Proposed packetization

We define the code gain g of a BOP as the number of packets that contain only protection symbols. With the packetization of [5], g may not take all values in the set $\{0,\ldots,N-1\}$. For example, suppose that $N=5$ and $s_j=34$. Then only $g=3$ and $g=4$ are admissible. We propose to make the construction of a BOP more flexible by inserting a dummy information symbol at the end of an information packet if needed. In this way, we guarantee that g can take all values in $\{0,\ldots,N-1\}$. We do not transmit the dummy symbols as they can be inserted at the decoder automatically just to realize a rectangular BOP and hence do not contribute to the transmitted bit budget. For the above example, the values $g=0,1,2$ can be obtained with $(m_j=7,d_j=1)$, $(m_j=9,d_j=2)$, and $(m_j=12,d_j=2)$, respectively (Table 2). Here d_j denotes the total number of dummy symbols for the BOP. The N possible packetizations corresponding to the N values $g=0,\ldots,N-1$ can be obtained in a systematic way as follows. For a given code gain $g\in\{0,\ldots,N-1\}$, let $m_{j,g},d_{j,g}$, and $k_{j,g}$ denote the size of the packet, the number of dummy symbols, and the number of information packets, respectively. Then we have $k_{j,g}=N-g$ and $(m_{j,g},d_{j,g})$ is the unique pair solving the equation $k_{j,g}m_{j,g}=s_j+d_{j,g}$ subject to $0\le d_{j,g}\le k_{j,g}-1$. The number of protection symbols associated to a code gain g for the layer j is $c_{j,g}=gm_{j,g}$.

3.2. Proposed algorithm

Using the BOP packetization of Section 3.1, we provide an algorithm for finding an optimal number of transmitted layers $L \in \{1, \ldots, M\}$ and a corresponding optimal code gain allocation $\mathbf{g}_L = (g_1, \ldots, g_L)$ given a total transmission budget R. Note that once an optimal code gain allocation is known, one can derive immediately the corresponding optimal error protection allocation $\mathbf{c}_L = (c_{1,g_1}, \ldots, c_{L,g_L})$ as explained in Section 3.1.

Let $P(x) = \sum_{j=0}^{x} p_N(j)$, where $p_N(j)$ is the probability of losing j packets out of N transmitted ones. Let $\Delta E(g_i, \ldots, g_L)$ denote the expected reduction in the distortion when layers $i, \ldots, L$ are sent with code gains $g_i, \ldots, g_L$, respectively, given that layers $1, \ldots, i-1$ are correctly decoded. Then

$$\Delta E(g_i, \ldots, g_L) = \sum_{t=i}^{L} (E_{t-1} - E_t) \prod_{j=i}^{t} P(g_j).$$

Since the expected error at the receiver is

$$E(g_1, \ldots, g_L) = E_0 - \Delta E(g_1, \ldots, g_L)$$

an optimal code gain allocation should maximize the expression $\Delta E(g_1, \ldots, g_L)$. This can be done recursively with dynamic programming using the following result.

Lemma 1. *For all* $k = i+1, \ldots, L$

$$\begin{aligned}\Delta E(g_i, \ldots, g_L) &= \Delta E(g_i, \ldots, g_{k-1}) + \\ &\quad \Delta E(g_k, \ldots, g_L) \prod_{x=i}^{k-1} P(g_x).\end{aligned}$$

Proof. Let $\Delta E_t(g) = (E_{t-1} - E_t) P(g)$. Then

$$\begin{aligned}\Delta E(g_i, \ldots, g_L) &= \sum_{t=i}^{L} \Delta E_t(g_t) \prod_{j=i}^{t-1} P(g_j) \\ &= \sum_{t=i}^{k-1} \Delta E_t(g_t) \prod_{j=i}^{t-1} P(g_j) + \\ &\quad \prod_{x=i}^{k-1} P(g_x) \sum_{t=k}^{L} \Delta E_t(g_t) \prod_{j=k}^{t-1} P(g_j)\end{aligned}$$

which gives the desired result.

Let $P_i(r, g)$ be equal to $P(g)$ if $r \geq s_i + c_{i,g}$ and equal to zero, otherwise. Let $\Delta E_i(r, g)$ be equal to $\Delta E_i(g)$ if $r \geq s_i + c_{i,g}$ and equal to zero, otherwise. Let $\Delta G(r; i : L)$ denote the maximum expected reduction in distortion when layers $i, \ldots, L$ are sent over the channel with code gains $g_i, \ldots, g_L$, respectively, given that $\sum_{j=i}^{L} s_j + c_{j,g_j} \leq r$, and layers $1, \ldots, i-1$ were correctly decoded. That is,

$$\Delta G(r; i : L) = \max_{\sum_{j=i}^{L} s_j + c_{j,g_j} \leq r} \Delta E(g_i, \ldots, g_L).$$

Then Lemma 1 with $k = i+1$ gives for $1 \leq i < L$

$$\begin{aligned}\Delta G(r; i : L) = \max_{0 \leq g \leq N-1} \{&\Delta E_i(r, g) + \\ &P_i(r, g) \Delta G(r - s_i - c_{i,g}; i+1 : L)\}\end{aligned} \tag{2}$$

On the other hand, we have

$$\Delta G(r; L : L) = \max_{0 \leq g \leq N-1} \Delta E_L(r, g). \tag{3}$$

In a preprocessing step, we compute for each layer $i = 1, \ldots, M$ and for each code gain $g = 0, \ldots, N-1$, the set of protection symbols $c_{i,g}$ (see Section 3.1). Then we build the $M \times (R+1)$ array $\Delta G(r; i : M)$ for $i = M, \ldots, 1$ and $r = 0, \ldots, R$, recursively using (2) and (3).

Now, let $g^*(\Delta G(r; i : L))$ be a function that returns the smallest g that achieves $\Delta G(r; i : L)$ in (2) (respectively (3)) when $\Delta G(r; i : L) > 0$ and let it be 0 otherwise. Then $g^*(\Delta G(r; i : L))$ is the optimal code gain g for the ith layer (and the corresponding $c_{i,g}$ is the optimal number of protection symbols) given that r is the number of available transmission symbols for layers i to L, and that layers 1 to $i-1$ are already decoded.

Algorithm 1 summarizes our approach. The time complexity of the algorithm is $O(NMR)$, and its space complexity is $O(MR)$. One can speed up Algorithm 1 by noting that for each i, the array entry $\Delta G(r; i : M)$ need only be computed for $r = s_i$ to $R - \sum_{j=1}^{i-1} s_j$ as all other array entries are either 0 or will never be required.

Note that the algorithm can also be used when the size of a channel packet is constrained to be smaller than a given maximum length. In this situation, we simply include this constraint in the preprocessing step.

Algorithm 1 Optimal solution

```
Input: $M, s_1, \ldots, s_M, E_0, \ldots, E_M, N, R$.
Output: $L, g_1, \ldots, g_L$.
for $i = M$ to 1 do
  for $r = 0$ to $R$ do
    Compute $\Delta G(r; i : M)$ and $g^*(\Delta G(r; i : M))$
  end for
end for
Set $r = R$ and $L = 0$
for $i = 1$ to $M$ do
  if $r \geq s_i$ then
    $g_i = g^*(\Delta G(r; i : M))$
    $r = r - (s_i + c_{i,g_i})$ and $L = i$
  end if
end for
```

We can also efficiently compute an optimal solution to the optimization problem under the monotonicity constraint $g_{i+1} \leq g_i$. This will allow us to evaluate the performance loss due to this constraint.

For $0 \leq r \leq R$, $1 \leq i \leq L$, and $0 \leq g \leq N-1$, let $H^*(r, g; i : L)$ denote the maximum reduction in expected distortion when layers $i, \ldots, L$ are transmitted with code gains $g_i, \ldots, g_L$, respectively, given that $\sum_{j=i}^{L} s_j + c_{j,g_j} \leq r$, $g_i \leq g$, and layers $1, \ldots, i-1$ were correctly decoded. Then

$$\begin{aligned}H^*(r, g; i : L) = \max_{0 \leq x \leq g} \{&\Delta E_i(r, x) + \\ &P_i(r, x) \Delta G(r - s_i - c_{i,x}; i+1 : L)\}\end{aligned}$$

In this way, $H^*(R, N-1; 1 : L)$ is the maximum reduction in expected distortion when layers $1, \ldots, L$ are transmitted with code gains $g_1, \ldots, g_L$, respectively, given that $\sum_{j=1}^{L} s_j + c_{j,g_j} \leq R$ and $g_1 \leq N-1$. To compute $H^*(R, N-1; 1:L)$, we use the following relations. For $0 \leq r \leq R,\ 1 \leq i < L$, and $0 \leq g \leq N-1$, we have

$$\begin{aligned} H^*(r, g; i:L) = \max\{&\Delta E_i(r,g) + P_i(r,g) \\ &H^*(r - s_i - c_{i,g}, g; i+1:L), \\ &H^*(r, g-1; i:L)\} \end{aligned} \tag{4}$$

For $0 \leq r \leq R,\ 0 \leq g \leq N-1$, and $i = L$,

$$H^*(r, g; i:L) = \max_{0 \leq x \leq g} \{\Delta E_i(r, x)\}. \tag{5}$$

Exploiting (4), (5), and the flexible BOP packetization, an optimal solution that satisfies the constraint $g_{i+1} \leq g_i$ can be computed in $O(NMR)$ time (see [8] for a detailed description). Similarly, we can also efficiently compute an optimal solution to the optimization problem subject to the monotonicity constraint $c_{i+1,g_{i+1}} \leq c_{i,g_i}$ (see [8]).

4. EXPERIMENTAL RESULTS

We present results for a Bunny model consisting of 5597 faces and 2907 vertices. We first compressed the model with the CPM coder [6] to produce one base mesh and 11 update layers. Then we compressed the base mesh with the Edge Breaker algorithm [9]. The resulting number of information symbols (here bytes) was $s_1 = 4269, s_2 = 117, s_3 = 144, s_4 = 185, s_5 = 240, s_6 = 293, s_7 = 415, s_8 = 624, s_9 = 892, s_{10} = 1284, s_{11} = 1805$, and $s_{12} = 2650$. The distortions (measured by the quadric error metric [10]) were $E_1 = 1695, E_2 = 973.64, E_3 = 721.47, E_4 = 292.18, E_5 = 202.30, E_6 = 138.18, E_7 = 84.63, E_8 = 42.08, E_9 = 21.54, E_{10} = 11.33, E_{11} = 4.76$, and $E_{12} = 0$. The value of E_0 is arbitrary and was set to $E_0 = 5000$. The packet erasure channel was modeled as a two-state Markov process with average burst error length 5. The number of channel packets per BOP was equal to $N = 100$. The total transmission budget was $R = \sum_{j=1}^{12} s_j = 12918$.

Figure 1 shows the expected distortion as a function of the average packet loss rate for three approaches. The first one, "Constrained C-bits", assumes as in [5] that $c_{i+1,g_{i+1}} \leq c_{i,g_i}$. The second one, "Constrained Code Gain", assumes that $g_{i+1} \leq g_i$. The third approach uses Algorithm 1, which does not make any assumptions.

The experiments show that the monotonicity constraint on the number of protection symbols can lead to a significant loss in performance. Indeed, although the individual symbols in layer i are more important than those in layer $i+1$, the protection of layer i may require fewer protection symbols if it has fewer information symbols. The monotonicity constraint on the code gains also harms the performance because when the number of available protection symbols is not enough to increase the code gain of layer i, then it cannot be used by layer $i+1$ if it violates the condition $g_{i+1} \leq g_i$. This can lead to a poor use of the available protection symbols.

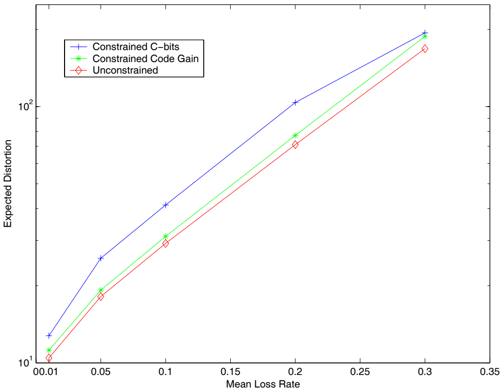

Fig. 1. Expected distortion as a function of the channel average packet loss rate for a Bunny model.

5. CONCLUSION

We proposed a flexible BOP packetization technique and an efficient source-channel allocation algorithm for the transmission system of [5]. Although it has linear time complexity, our algorithm cannot be used for real-time applications where the error protection allocation has to be computed online. However, we can use it to check the quality of suboptimal solutions, or in applications where the error protection is computed offline and used for a wide range of channel statistics.

Acknowledgment. We thank Renato Pajarola for providing us with the source code of the CPM algorithm. This work was supported by the DFG Research Training Group GK-1042.

6. REFERENCES

[1] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Proc. Mag.*, vol. 17, pp. 61–82, July 2000.

[2] R. Hamzaoui, V. Stanković, and Z. Xiong, "Optimized error protection of scalable image bitstreams," *IEEE Signal Proc. Mag.*, vol. 22, pp. 91–107, Nov. 2005.

[3] S. Bischoff and L. Kobbelt, "Streaming 3D geometry data over lossy communication channels," *Proc. IEEE ICME 2002*, Lausanne, Aug. 2002.

[4] Z. Yan, S. Kumar, and C.-C. J. Kuo, "Mesh segmentation schemes for error resilient coding of 3-D graphic models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 138–144, Jan. 2005.

[5] G. Al-Regib, Y. Altunbasak, and J. Rossignac, "Error-resilient transmission of 3-D models," *ACM Trans. Graphics*, pp. 182–208, April 2005.

[6] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Trans. Visual. Comput. Graphics*, vol. 6, no. 1, pp. 79–93, 2000.

[7] S. Dumitrescu and X. Wu, "Globally optimal uneven erasure-protected multi-group packetization of scalable codes," *Proc. IEEE ICME 2005*, pp. 900–903, Amsterdam, July 2005.

[8] S. Ahmad and R. Hamzaoui, "Optimal error protection of progressively compressed 3D meshes", *Konstanzer Schriften in Mathematik und Informatik* [Online], `http://www.inf.uni-konstanz.de/Preprints/preprints-all.html`.

[9] H. Lopes, J. Rossignac, A. Safonova, A. Szymczak, and G. Tavares, "Edgebreaker: A simple compression algorithm for surfaces with handles," *Computers & Graphics*, vol. 27, no. 4, pp. 553–567, 2003.

[10] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," *Proc. SIGGRAPH 97*, pp. 209–216, 1997.