A JOINT SOURCE AND CHANNEL CODING APPROACH FOR PROGRESSIVELY COMPRESSED 3-D MESH TRANSMISSION

Ghassan Al-Regib, Yucel Altunbasak* and Jarek Rossignac***

* *Center for Signal and Image Processing*
Georgia Institute of Technology
Atlanta, GA 30332-0250
E-mail:{gregib, yucel} @ece.gatech.edu

** *Graphics, Visualization and Usability Center*
Georgia Institute of Technology
Atlanta, GA 30332-0280
E-mail: jarek@cc.gatech.edu

ABSTRACT

In this paper, we present an unequal error protection method for packet-loss resilient transmission of progressively compressed 3-D meshes. The proposed method is based on a source and channel coding approach where we set up a theoretical framework for the overall system by which the channel packet loss behavior and the channel bandwidth can be directly related to the decoded mesh quality at the receiver. In particular, we develop a statistical distortion measure and optimize it to compute the best[1] combination of (i) the number of triangles to transmit, (ii) the total number of channel coding bits, and (iii) the distribution of these error-protection bits among the transmitted layers in order to maximize the expected decoded mesh quality at the receiver.

The proposed method differs from the earlier approaches [1] in two major aspects: (i) determination of the number of channel coding bits (C) and (ii) the approach of reducing the source rate in order to accommodate for channel coding bits. In [1], C is assumed to be given, whereas the proposed method optimally computes C. Also, the approach in [1] uses coarser quantizers to reduce the number of source coding bits while here we send fewer triangles and keep the geometry precision fixed. When the proposed method is used to transmit a typical 3-D mesh over a channel with a 10% packet loss rate, the distortion (measured using the *Hausdorff* distance between the original and the decoded overly-sampled meshes) is reduced by 50% compared to the case when no error protection is applied.

1. INTRODUCTION

An increasing number of Internet applications exploit highly detailed 3-D meshes, giving rise to a large number of data bits to be stored, transmitted, and rendered within a limited time frame. For example, a typical 3-D polygonal mesh of 60,000 triangles requires 5.5 seconds of transmission time over 1.544Mbps T1 line assuming that we use three vertex indices per triangle and 10 bits per vertex coordinate. This latency prohibits smooth navigation within a networked virtual environment. To alleviate such limitations, the mesh can be compressed using a single-layer compression technique [2, 3]. Nevertheless, single-level compression techniques hinder the receiver from rendering and displaying the mesh before the bit-stream is thoroughly downloaded. To reduce such latency, progressive compression techniques have been designed so that a coarse mesh is sent first to be displayed on the client's screen and then refinement information is transmitted later to transform the crude mesh to a set of finer meshes till the full mesh is decoded and displayed on the client screen [4].

[1]Optimality should be understood within the context of the particular distortion measure and the FEC codes used in our approach.

Even though all compression techniques that exist in the literature reduce both the required bandwidth and the latency, they do not address a major factor that affects the decoded mesh quality, which is channel packet losses. Approaches to recover from such losses can be *network-oriented solutions* such as TCP, *post-processing solutions* such as error concealment, or *pre-processing solutions* such as forward error correction (FEC) codes. In this paper, we only consider FEC based pre-processing approaches. Generally speaking, pre-processing methods fall into two classes: equal error-protection (EEP) and unequal error-protection (UEP). EEP methods apply the same FEC code to all parts of the bit-stream regardless of the contribution of each part to the decoded mesh quality. EEP is suitable when the channel has a low packet loss rate. However, at higher packet loss rates, considerable degradation in the decoded mesh quality is expected because of the high probability that important parts of the bit-stream will get lost. In this case, UEP is more suitable since important parts get higher level of error-protection than other parts.

The two unequal error-protection (UEP) methods, the one proposed in [1] and the one proposed in this paper, differ in two major aspects. In the latter method the total number of channel coding bits (C) is determined optimally which is not the case in the former method. The second major difference is that the proposed method in this paper accommodates for channel coding bits by sending fewer triangles while keeping the same geometry position precision while the method proposed in [1] accommodates for channel coding bits by using coarser quantizers. More specifically, the proposed method optimally computes both the total number of channel coding bits and the distribution of these bits among the transmitted layers[1]. To this effect, we develop a statistical distortion measure that estimates the distortion at the decoded mesh. Then, we jointly design both source and channel codes to minimize the expected distortion for a given channel.

We compress 3-D meshes using the *Compressed Progressive Mesh* (CPM) [4] algorithm. At the encoder, CPM applies a series of *edge-collapse* operations while at the decoder it applies a series of *vertex-split* operations. CPM produces M batches in addition to the base-mesh. Application of each batch yields a different level of detail (LOD) that further approximates the original 3-D mesh. The base-mesh can be compressed using any single-level mesh compression technique [2, 3].

In our work, we adapt a packetization method known as block of packets (BOP) [5]. In this method, the data is placed in horizontal packets and FEC is applied vertically. Then the packets are transmitted horizontally. Let (n, k_l) be the block FEC code applied to a given BOP. Then, if the number of lost packets is not more than $(n - k_l)$, then the decoder will be able to recover all lost packets in this BOP. Otherwise, the decoder considers these packets as irrecoverable and the decoding process is terminated. In our UEP implementation, every encoded level is packetized into one BOP. Each BOP is protected with an optimal[1] FEC code rate that is derived to maximize the expected decoded mesh quality at the receiver as detailed in Section 2. The forward error correction codes used in this paper are the Reed-Solomon (RS) codes that are maximal distance separable codes.

2. PROPOSED UNEQUAL ERROR-PROTECTION METHOD

In this section, we develop a statistical distortion measure that has the channel bandwidth and the channel error characteristics as parameters. Then, we minimize this distortion function with respect to the number of channel coding bits (C), and [$C^{(0)}$, $C^{(2)}$, ..., $C^{(L)}$], where $C^{(j)}$ is the number of error-protection bits to be assigned for the j^{th} level, and $L \leq M$ (M is the total number of encoded levels while L is the number of transmitted levels).

2.1. Statistical Distortion Measure

The total bit rate should be kept the same by reducing the number of source coding bits to accommodate for the RS redundant bits. For a progressively compressed 3-D mesh, the source bits can be reduced by either (i) using coarser quantizers to reduce the geometry information precision, or (ii) reducing the number of polygons to be sent while keeping the same precision of the transmitted geometry information. We choose the second method where error-protection bits are accommodated by reducing the number of transmitted polygons.

Generally speaking, mesh distortion estimation depends on the decoding strategy. CPM decoding process terminates whenever lost packets cannot be recovered since the loss of connectivity information results in synchronization problems at the decoder. Following this standard decoding strategy, the expected distortion at the received mesh is the sum of the products $P_j E_j$, where j is the level index, E_j is the distortion that would result when the decoding process stops at level-j, and P_j is the probability of having an irrecoverable packet loss at level-j. There are two ways to estimate the distortion (*i.e.*, E_j) and these two methods are discussed in section 2.2. P_j, the probability of discontinuing the decoding operation at level-j, is a conditional probability. It is equal to the products of the probabilities of correctly decoding all data before level-j, but not being able to decode the j^{th} level. In equation form, the expected mesh distortion at the decoder is written as follows:

$$D_r(L) = P_0 E_0 + \sum_{j=1}^{L-1} P_j E_j \prod_{k=0}^{j-1} (1 - P_k) + \prod_{k=0}^{L} (1 - P_k) E_L \qquad (1)$$

where L is the total number of transmitted levels and it is limited to $0 < L \leq M$. P_j is the probability of having irrecoverable packets in the j^{th} level, and E_j is the error between the transmitted mesh and the mesh after decoding level-$j - 1$.

In Equation 1, P_j can also be defined as the probability of losing more than $n - k_j$ packets in the j^{th} layer, since RS-code (n, k_j) applied at this layer will recover fewer number of packet losses. This quantity can be calculated as follows:

$$P_j = \sum_{m=n-k_j+1}^{n} p(m, n) \qquad (2)$$

where $p(m, n)$ denotes the block error density function, *i.e.*, the probability of losing m symbols within a block of n symbols. $p(m, n)$ depends on the channel model. In here, we used the Gilbert-Elliot channel model and the reader is referred to [5] for more details on calculating $p(m, n)$ for this two-state Markov model. Calculating the estimated error (E_j) is discussed in the following section.

2.2. Determining the Rate-Distortion Curve (E_j's)

There are two methods to estimate the error introduced on the mesh by terminating the decoding process at a given LOD (*i.e.*, E_j in Equation 1). A natural metric is to compute the actual error between the transmitted mesh and the resulting mesh produced by a given LOD. One way to do so is to measure the *Hausdorff* distance that estimates the maximum error between the two meshes. However, calculating the *Hausdorff* distance is an expensive operation and it requires considerable processing power as well as memory space. Therefore, we propose to use a less-expensive metric to produce a similar plot that reflects the *relative* error between different layers of the bit-stream. We use the sum of the square distances between each vertex V of the simplified mesh and the planes that support the original triangles that were incident upon all the vertices that collapsed to V. This metric is a *Quadric Error Metric* and it is in fact used in choosing the edges to be collapsed at every iteration in the encoding process. Figure 1 depicts the estimated error (using the *Quadric Error Metric*) between the transmitted SMALL BUNNY mesh and each of the 10 LODs produced by CPM. As shown in the plot, as the number of received bits increases, the error between the transmitted and the decoded meshes decreases.

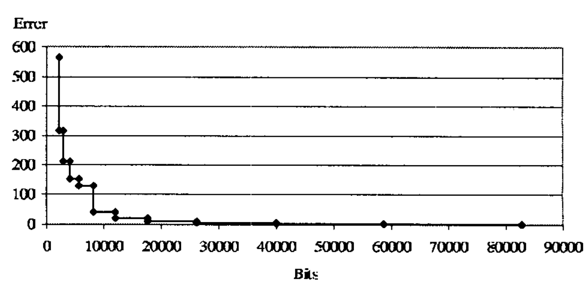


Fig. 1. The rate-distortion (R-D) curve for the SMALL BUNNY mesh using the *Quadric Error Metric.*

2.3. Solution to the Optimization Problem

Equation 1 estimates the expected distortion introduced at the decoded mesh in a statistical sense. Now, the objective is to (i) choose an optimal error-protection bit budget (C), and (ii) optimally distribute these C bits among the transmitted L levels (*i.e.*, to choose L, C^0, C^1, ... , and C^{L-1}) in order to minimize the expected distortion at the decoded mesh. We *jointly* compute the optimal values for these quantities by minimizing the distortion in Equation 1. Intuitively, the base-mesh is usually regarded as the most important layer in the encoded bit-stream, followed by the coarsest LOD, and so on, till the finest LOD. Therefore, we expect that the optimization process to allocate more error-protection bits to the base-mesh and the first few coarse layers.

The error-protection bit budget is upper-bounded by C, the maximum number of available error-protection bits. Combining Equations 1 and 2 together with the above condition results in a constrained optimization problem given as follows:

$$\min_{C,C^{(j)}} arg\ P_0E_0 + \sum_{j=1}^{L-1} P_jE_j \prod_{k=0}^{j-1} (1-P_k) + \prod_{k=0}^{L} (1-P_k)E_L \quad (3)$$

$$subject\ to: \sum_{j=0}^{L} C^j = C, \quad j = 0, ..., L \quad (4)$$

The solution of the above constrained optimization problem consists of two parts: C and the vector $\mathbf{C_L} = [C^{(0)}, C^{(1)}, ..., C^{(L-1)}]$, where $L \leq M$. To solve this problem we develop a local search hill-climbing algorithm that makes assumptions about the data but computationally tractable. An essential assumption is that the number of error-protection bits assigned to a given level should be less than the number of error-protection bits assigned to a coarser level (*i.e.*, $C^{(L-1)} \leq ... \leq C^{(1)} \leq C^{(0)}$, where $L \leq M$).

For a given packet loss rate (P_{LR}) and a given total bit rate, the optimal solution is found via an iterative algorithm. In each iteration C is set to a certain value and a local search algorithm is run to determine the vector $\mathbf{C}_L$ that minimizes the expected distortion (*i.e.*, D_r). Since our solution is a batch-by-batch, C can take only certain values. These values are from the set { 0, $S^{(M)}$, $S^{(M)}+S^{(M-1)}$, ..., $\sum_{j=1}^{M} S^{(j)}$ }, where $S^{(j)}$ is the number of bits in the j^{th} level. In other words, in the first iteration, C is set to zero bits, in this case all M levels are transmitted (*i.e.*, $L = M$), and the corresponding distortion is calculated. In the second iteration, C is set to $S^{(M)}$, in this case $L = M - 1$ levels are transmitted, and the local search algorithm is run to determine the optimal $\mathbf{C}_L$ and the corresponding distortion is calculated. The same process is repeated M times and the C that gives the minimum distortion is considered as the optimal C and the corresponding vector $\mathbf{C}_L$ is considered as the optimal distribution of error-protection bits.

Applying this algorithm to the 10-level progressively compressed SMALL BUNNY mesh (consisting of 9580 triangles) results in the curves shown in Figure 2. Each curve depicts the expected distortion for a given packet loss rate and a spectrum of values of the error-protection bit budget (C). The three curves correspond to the three packet loss rates (P_{LR}) 0.0, 0.12, and 0.4, respectively. As shown in the plot, when the packet loss rate is zero, the optimization algorithm assigns zero bits for error-protection. However, when the packet loss rate increases to 0.12 and 0.40, the optimum error-protection bit-budget is about 3700 and 7100 bytes, respectively. In all these cases, 10348 bytes are transmitted. The number of transmitted levels (out of the encoded 11 levels) in these three cases turned out to be 11, 10, and 8, respectively.

Even though the optimization algorithm adds to the complexity of the proposed system, it considerably improves the whole system performance as will be shown in the following section. In reality, several 3-D applications do not require online encoding of 3-D meshes and hence we can perform all calculations off-line. For example, we might store all meshes in the CPM format together with a pre-computed look-up table that lists for every packet loss rate the quantities L, C, and $\mathbf{C_L}$ in the server. Whenever a client asks for a mesh (or more), the server calculates the aforementioned parameters according to the channel characteristics and streams out the optimally encoded and protected 3-D mesh.

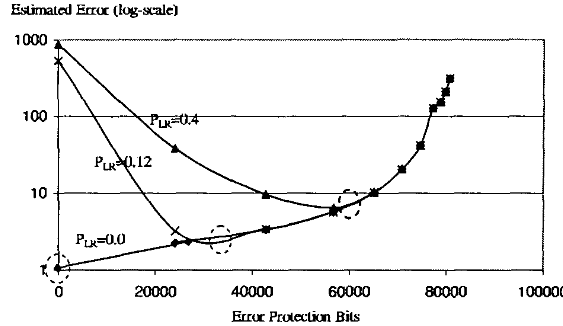


Fig. 2. The estimated distortion at the decoded SMALL BUNNY mesh as a function of the error-protection bit-budget (C). The small circles indicate the optimum C for each of the three cases.

3. SIMULATION RESULTS

To demonstrate the efficacy of the proposed UEP method, we used both subjective and objective methods of comparison. In particular, we used the *Hausdorff* distance $\mathcal{D}$ between densely sampled points on the original (transmitted mesh) and the decoded mesh as an objective comparison metric. In the following experiments, the total bit budget is assumed to be given and is kept the same for all three methods (*i.e.*, no error-protection (NEP), equal error-protection (EEP) and unequal error-protection (UEP)) for fair comparison.

We applied the proposed joint source and channel coding method on the SMALL BUNNY 3-D mesh shown in Figure 4(a). This mesh has 5960 vertices, 9580 faces and is compressed progressively into 20 batches using the CPM algorithm and the resulting encoded bit-stream contains 72429 bits. In all experiments the average burst length, L_B, is set to 5 and the RS packet size is set to 100 (*i.e.*, $n = 100$).

Figure 3 depicts $\mathcal{D}$ as a function of the packet loss rate P_{LR} for the SMALL BUNNY mesh. Three curves in this figure represent the cases of EEP, UEP, and NEP, respectively. As can be seen from these curves, for an error-free channel, no bits are assigned for channel coding and hence the distortion is zero. Note that we used fine quantizers for geometry information and hence the transmitted and the original meshes are identical. As the packet loss rate (P_{LR}) increases, EEP and NEP performances become closer to each other since lost packets of coarse levels are irrecoverable. On the other hand, the proposed UEP method manages to protect these layers by assigning them more error-protection bits and therefore, the degradation in the decoded mesh quality is more graceful compared to the degradation in the other two methods. More specifically, when $P_{LR} \geq 18\%$, the base-mesh packets get lost and only UEP can recover these lost packets as can be seen from the curves in Figure 3. The optimal distribution of the total

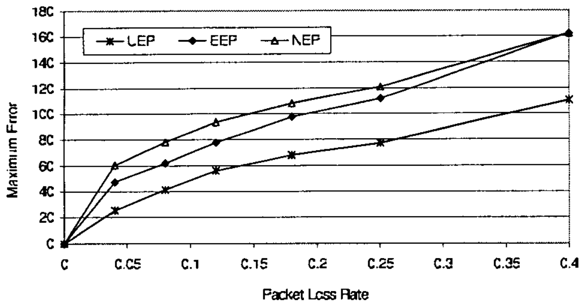


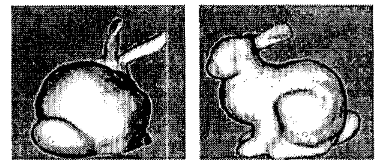
Fig. 3. Maximum Error (*Hausdorff* distance) between the transmitted and the decoded SMALL BUNNY meshes.

bit budget between source and channel coding bits is tabulated in Table 1.

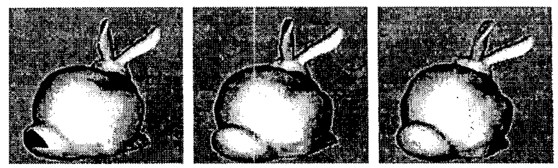
P_{LR}	# transmitted levels	C	S
0.00	20	0	72429
0.04	18	11485	60944
0.08	16	18566	53863
0.12	16	18566	53863
0.18	14	25366	47063
0.25	12	31620	40809

Table 1. The channel coding bits (C) versus source coding bits (S) for a spectrum of packet loss rates for the SMALL BUNNY mesh.

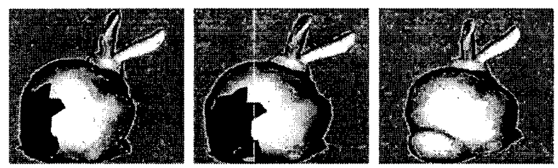
Subjective results are shown in Figure 4 for the SMALL BUNNY mesh. In these results, we had to control the location of the packet loss in order to show the mesh from the same side in all cases. Note how the UEP method protects the base-mesh and few other coarse batches even when the packet loss rate is 40%. Moreover, note how the UEP method kept a reasonable level of detail at all packet loss rates compared to the other two methods (NEP and EEP).



(a) Original SMALL BUNNY mesh



(b) NEP ($P_{LR} = 0.08$) (c) EEP ($P_{LR} = 0.08$) (d) UEP ($P_{LR} = 0.08$)



(e) NEP ($P_{LR} = 0.40$) (f) EEP ($P_{LR} = 0.40$) (g) UEP ($P_{LR} = 0.40$)

Fig. 4. Subjective results of applying NEP, EEP and UEP methods on the SMALL BUNNY mesh.

4. CONCLUSIONS

In this paper, we presented an error-resilient method for 3-D mesh transmission. The proposed method is scalable with respect to both channel bandwidth and channel error characteristics. The bit budget allocation method (i) assigns optimal error-protection bit budget (C), and (ii) distributes these error-protection bits among the transmitted layers to maximize the decoded mesh quality. These optimal RS codes depend on: the error-protection bit budget, the channel packet loss rate, and batch-by-batch rate-distortion characteristics of the source mesh. Moreover, in order to keep the bit rate unchanged when error-protection bits are added, we reduce the source coding bits by reducing the number of polygons transmitted to the client, which differs from the method used in [1] where coarser quantizers were used while all batches were transmitted. The authors currently study the effect of combining these two approaches of reducing the number of source coding bits on the whole system performance.

Experimental results show that with our UEP approach, the quality of the decoded mesh degrades more gracefully as the packet loss rate increases. The applicability of the proposed UEP method does not depend on a particular 3-D mesh progressive compression algorithm, although we used CPM in this paper. Finally, the applicability of the proposed UEP method does not depend on a particular channel model, although we used Gilbert-Elliot model in this paper.

Acknowledgments

This work was supported in part by Georgia Tech Broadband Institute (GTBI).

5. REFERENCES

[1] Ghassan Al-Regib, Yucel Altunbasak, and Jarek Rossignac, "An unequal error protection method for progressively compressed 3-D meshes," in *The Proceedings of the IEEE ICASSP'02 (Accepted for Publication)*, Orlando, FL, May 2002.

[2] J. Rossignac, "3-D compression," in *Lecture at the ACM SIGGRAPH'99*, 1999.

[3] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proceedings of Graphics Interface*, Vancouver, Canada, June 1998.

[4] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 79–93, January-March 2000.

[5] U. Horn, K. Stuhlmuller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communications*, vol. 15, pp. 77–94, 1999.