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 (UEP) 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 combination of (i) the number of triangles to transmit, (ii) the total number of channel coding bits, and (iii) the distribution of the correction coding bits among the transmitted layers in order to maximize the 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) source rate reduction approach to accommodate for the channel coding bits. In [1], C is assumed to be given, whereas the proposed method computes the optimal C. Also, the approach in [1] uses coarser quantizers to reduce the number of source coding bits while here we send fewer triangles while keeping the geometry precision fixed. Experimental results show that with the proposed unequal error protection (UEP) approach, the decoded mesh quality degrades more gracefully (compared to either no error protection (NEP) or equal error protection (EEP) methods) as the packet loss rate increases.

1. BACKGROUND

An increasing number of Internet applications utilize highly detailed 3-D meshes, giving rise to a large amount of data 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 even over 1.54Mbps T1 line assuming each triangle is represented by 18 bytes. This latency prohibits a smooth navigation within a networked virtual environment. To alleviate such limitations, the mesh can be compressed using one of the several single-layer compression techniques that are summarized in [2]. However, the resulting compressed bit-stream still requires a significant amount of time to be completely downloaded before it can be decoded and displayed on the client’s screen. To reduce latency, progressive compression techniques have been designed so that a coarse mesh is sent first and then refinement information is transmitted later to transform the received crude mesh to a set of finer meshes till the full mesh is decoded on the client side [3].

Even though all compression techniques that exist in the literature reduce both the required bandwidth and the latency, they do not address other major factors that affect the decoded mesh quality. Among these factors is the channel packet loss. Approaches to recover from such losses can be: network-oriented solutions such as TCP, post-processing solutions, or pre-processing solutions. 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, important parts of the bit-stream might be lost, which results in a considerable degradation in the decoded mesh quality. In this case, UEP is more suitable since important parts of the bit-stream 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. First, in the former method the total number of channel coding bits (C) was not determined optimally while in the proposed method in this paper C is determined optimally. 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 computes optimally both the total number of channel coding bits and the distribution of these bits among the transmitted layers. To this effect, we develop a statistical distortion measure that estimates the distortion at the decoded mesh. Then, we design both source and channel codes jointly to minimize the expected distortion for a given channel.

We compress 3-D meshes using the Compressed Progressive Mesh (CPM) [3] 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. In this paper, the base-mesh is compressed using the TG-algorithm [4].

In our work, we adapt a packetization method known as block of packets (BOP) [5]. In this method, the data is placed in horizon-
tal packets and then FEC is applied horizontally. Then the packets are transmitted vertically. Let \((n, k_i)\) be the RS code applied to a given BOP. Then, if the number of lost packets is not more than \((n - k_i)\), 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, the base-mesh bit-stream is packetized into one BOP while every LOD’s batch is packetized into one BOP. Hence, for \(M\)-level encoded bit-stream, there are \(M + 1\) BOPs. Each BOP is protected with an optimal 1 FEC code that is derived to maximize the decoded mesh quality at the receiver as detailed in Section 2. The forward error correction (FEC) 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 source coding bits \((C)\), and \(C^{(0)}, C^{(2)}, \ldots, C^{(L)}\), where \(C^{(j)}\) is the number of error protection bits to be assigned for the \(j\)th batch, and \(L \leq M\) is the number of batches sent to the client, in order to maximize the decoded mesh quality.

2.1. Statistical Distortion Measure

In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio. In any error protection technique, the total bit rate should be kept the same by reducing the number of source bits by the same ratio.

\[
P_j = \sum_{m=n-k_j+1}^{n} p(m, n)
\]

where \(p(m, n)\) denotes the block error density function, \(i.e.,\) the probability of losing \(m\) symbols within a block of \(n\) symbols.

So far, only two quantities in Equations 1 and 2 have not been calculated yet. One is the block error density function, \(p(m, n)\), which depends on the channel mesh used. In here, we used the G-E model and the reader is referred to [5] for more details on calculating \(p(m, n)\) for this channel mesh. The other quantity is the estimated error \((E_j)\), which 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 the given LOD. One way to do so is to measure the Hausdorff distance that estimates the maximum error between the two meshes.

Computing the Hausdorff distance results in a rate-distortion curve that gives the distortion at the decoded mesh as a function of the number of received bits. 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 maximum of the 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 known as 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 maximum error between the transmitted mesh and the decoded one decreases.

![Fig. 1. The rate distortion (R-D) curve for the small bunny mesh using the Quadric Error Metric.](image-url)
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 \(i.e., C\), and (ii) optimally distribute \(C\) bits among the transmitted \(L\) levels \(i.e., to\ choose \(L\) and \(C^0, C^1, \ldots, C^L\) in order to minimize the distortion at the decoded mesh. We jointly compute the optimal values for these quantities using the statistical distortion measure shown 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 \(L + 1\) quantities, \(k_j\), in Equation 1 must satisfy two main conditions. First, the error protection bit budget is upper-bounded by \(C\), the maximum number of available error protection bits. The second and more-obvious constraint is that \(k_j\) cannot be greater than \(n\). Combining Equations 1 and 2 together with the above two conditions results in a constrained optimization problem given as follows:

\[
\min \arg_{k_j,C} \sum_{m=0}^{n} \frac{p(m,n)}{x} E_0 + \sum_{j=0}^{L} \sum_{k_j=0}^{n} p(m,n) x_j + \sum_{j=0}^{L} (1 - \sum_{m=0}^{n} p(m,n)) x_j + \sum_{j=0}^{L} (1 - \sum_{m=0}^{n} p(m,n)) x_{L+1} \\
\text{subject to:} \sum_{j=0}^{L} C^j = C, \quad 0 \leq k_j \leq n, \quad j = 0, \ldots, L
\]

(3)

The first constraint in Equation 4 forces the solution to be scalable with respect to the bit budget. However, the second constraint in Equation 4 is a natural constraint of any FEC block code. Similarly, incorporating the block error density function in Equation 3 forces the solution to be scalable with respect to channel error characteristics.

The solution of the above constrained optimization problem consists of two parts: \(C\) and the vector \(C_L = [C^{0}, C^{1}, \ldots, C^{L}]\), where \(L \leq M\) and \(\sum_{j=0}^{L} C^j = C\). 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 \(i.e., C^j \leq C^{j+1} \leq \ldots \leq C^L \leq C^0\) where \(L \leq M\).

For a given packet loss rate \(P_L\) 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 \(C_L\) that minimizes the expected distortion \(i.e., D_C\). Since our solution is a batch-by-batch, \(C\) can takes only certain values. These values are from the set \{ 0, \(S^M\), \(S^{(M)} - S^{(M-1)}, \ldots, \sum_{j=0}^{j} S^{(j)} \}\), where \(S^{(j)}\) is the number of bits in the \(j^{th}\) batch. In other words, in the first iteration, \(C\) is set to zero bits, in this case all \(M\) levels are transmitted, and the corresponding distortion is calculated. In the second iteration, \(C\) is set to \(S^M\), in this case \(M - 1\) levels are transmitted, and the local search algorithm is run to determine the optimal \(C_L\) 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 \(C_L\) 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 \(P_L\) 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 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.

![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.](image)

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, many 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, used in a video game, in the CPM format together with a pre-computed look-up table that lists for every packet loss rate the quantities \(L, C\), and \(C_L\) in the game server. Whenever a client (player) joins the game and chooses 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.

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 \(D\) 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 \(i.e., no error protection, equal error protection and unequal error protection\) for fair comparison.

We applied the proposed unequal error protection (UEP) 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 FEC packet size is set to 100 (i.e., $n = 100$).

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

Figure 3 depicts $D$ as a function of the packet loss rate $P_B$ 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 ($P_B$) increases, EEP and NEP performances become closer to each other since the lost packets of the coarse levels are irrecoverable. On the other hand, UEP managed to protect these layers by assigning 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 $P_B \geq 18\%$, 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 bit budget between source and channel coding bits is tabulated in Table 1 (Footnote 1).

![Fig. 3. Maximum Error (Hausdorff distance) between the transmitted and the decoded SMALL BUNNY meshes.](image)

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

<table>
<thead>
<tr>
<th>$P_B$</th>
<th># transmitted batches</th>
<th>$C$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>20</td>
<td>0</td>
<td>72429</td>
</tr>
<tr>
<td>0.04</td>
<td>18</td>
<td>11485</td>
<td>60944</td>
</tr>
<tr>
<td>0.08</td>
<td>16</td>
<td>18566</td>
<td>53863</td>
</tr>
<tr>
<td>0.12</td>
<td>16</td>
<td>18566</td>
<td>53863</td>
</tr>
<tr>
<td>0.18</td>
<td>14</td>
<td>25366</td>
<td>47063</td>
</tr>
<tr>
<td>0.25</td>
<td>12</td>
<td>31620</td>
<td>40809</td>
</tr>
</tbody>
</table>

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).

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 are used while all batches are 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. Finally, the applicability of the proposed UEP method does not depend on a particular 3-D mesh compression algorithm, although we used CPM in this paper. Moreover, the applicability of the proposed UEP method does not depend on a particular channel model, although we used G-E model in this paper.

5. REFERENCES


