Bit Allocation for Joint Source and Channel Coding of Progressively Compressed 3-D Models

Ghassan Al-Regib, Yucel Altunbasak, *Senior Member, IEEE*, and Russell M. Mersereau, *Fellow, IEEE*

***Abstract*—This paper presents a joint source and channel coding method for transmission of progressively compressed three-dimensional (3-D) models where the bit budget is allocated optimally. The proposed system uses the *Compressed Progressive Mesh (CPM)* algorithm to produce a hierarchical bitstream representing different levels of detail (LODs). In addition to optimizing the transmitted bitstream with respect to the channel characteristics, we also choose the optimum combination of quantization and tessellation to maximize the expected decoded model quality. That is, given a 3-D mesh, a total bit budget (B) and a channel packet-loss rate (P_{LR}), we determine the optimum combination of: 1) the geometry coordinates quantizer step size (l-bit); 2) the number of transmitted batches (L); 3) the total number of channel coding bits (C); and 4) the distribution of these channel coding bits among the transmitted levels ($\mathrm{C_L}$). Experimental results show that, with our unequal error protection approach that uses the proposed bit-allocation algorithm, the decoded model quality degrades more gracefully (compared to either no error protection or equal error protection methods) as the packet-loss rate increases.**

***Index Terms*—Bit allocation, joint source and channel coding, packet-loss resilient transmission, streaming, three-dimensional (3-D) graphics, 3-D graphics streaming, 3-D mesh compression, unequal error protection (UEP), virtual reality over IP networks.**

I. Introduction

INTERACTIVE graphics applications use three-dimensional (3-D) meshes intensively. These 3-D meshes represent such computer models of 3-D objects as manufacturing assemblies, geographical databases, virtual environments for e-commerce applications, or entertainment applications. It is expected that Internet-based access to these 3-D models will soon be common in fundamental areas of manufacturing, architecture, petroleum exploration, urban planning, tourism, defense, medicine, e-commerce, and entertainment. All of these applications require access to large 3-D models over bandwidth-limited links. This creates a bottleneck. For example, a typical 3-D triangular mesh of 60 000 faces requires 5.5 s of transmission time even over a 1.544-Mb/s T1 line assuming that each triangle is represented by 18 bytes. A related issue is the required storage space at the client machine.

To alleviate such limitations, one might apply a single-level 3-D mesh compression algorithm [1]–[18] to reduce the transmitted data size. However, in this case, the encoded bitstream is not scalable. As a result, low-capability clients will suffer from a large delay. Moreover, in a networked virtual reality application such as video gaming, high-capability clients will be subject to latency caused by the low-capability clients. This deters the smooth navigation required by most 3-D graphics applications.

To address this issue, 3-D graphics community researchers have begun to develop scalable 3-D mesh compression algorithms [19]–[24]. These algorithms are designed with two constraints in mind; namely, bandwidth limitation and client machine capability. The common idea underlying all of these methods is to send a coarse representation of the mesh followed by enhancement bits that refine the mesh to its finest level. The transmission (and equivalently the decoding) process can be terminated at any point when either the available bit budget has been consumed or the available storage space is fully exploited. Even though these scalable compression algorithms improve bandwidth utilization, they do not address other major network considerations such as packet losses. Because of the sensitivity and the interdependence of the encoded bitstream levels, when a packet is lost, all following packets will be discarded [20], [25]. As a result, packet losses result in catastrophic distortion in the decoded 3-D model. Therefore, error-resilient preprocessing and/or postprocessing needs to be developed to reduce the end-to-end delay and to increase the decoded model quality while keeping the bandwidth unchanged.

The effectiveness of preprocessing methods versus postprocessing methods depends on the application as well as the applied method.[1] In this paper, we consider forward error correction (FEC)-based preprocessing approaches since they do not require extra processing on the client side. Generally speaking, FEC preprocessing 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 bitstream regardless of the contribution of each part to the decoded model quality. EEP is suitable when the channel has a very low packet-loss rate. However, at higher packet-loss rates, it is helpful to protect various parts of the encoded bitstream according to their importance. Since losing important parts of the bitstream results in a considerable degradation in the decoded model quality, UEP is more suitable at these rates since important parts of the bitstream get a higher level of error protection than other parts.

Both the efficiency and the effectiveness of UEP methods depend on the criterion used to classify the encoded bitstream. Such classification may depend on source characteristics only

Manuscript received January 30, 2002; revised June 20, 2003. This paper was recommended by Associate Editor H. Watanabe.

G. Al-Regib is with the Center for Signal and Image Processing, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA and also with the Georgia Institute of Technology, Savannah, GA USA (e-mail: gregib@ece.gatech.edu).

Y. Altunbasak and R. M. Mersereau are with the Center for Signal and Image Processing, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: yucel@ece.gatech.edu; rmm@ece.gatech.edu).

Digital Object Identifier 10.1109/TCSVT.2004.841638

[1] For a comprehensive review of such preprocessing and postprocessing methods in the case of video communications, the reader is referred to [26].

[27], [28], channel characteristics only, or both source and channel characteristics. An error-resilient 3-D model coding method has been proposed by Yan *et al.* [28]. It is based on the constructive traversal compression scheme proposed by Li and Kuo [31]. In this scheme, the mesh is partitioned into several segments and each segment is transmitted independently. These segments are then stitched at the decoder using joint-boundary information. This error-resilient method classifies the joint-boundary information as the most important part of the bitstream. However, assigning error-protection bits to the different parts of the bitstream is achieved experimentally (i.e., no analytical approach has been provided for error-protection bit allocation). Although this pioneering work deserves credit, its application areas are limited since: 1) it is designed for a specific 3-D compression algorithm [31] and 2) it encodes each part of the mesh at full resolution instead of considering hierarchical encoding, which provides a bitstream that is scalable with respect to the channel bandwidth.

In this paper, we propose a UEP method that depends on both source and channel characteristics. In this case, the source and channel codes are jointly designed to maximize the decoded model quality. A major issue in designing bit-allocation algorithms is determining the number of bits needed to be assigned for both source data and channel codes. In order to keep the total number of transmitted bits the same, the number of source coding bits needs to be reduced to accommodate the channel coding bits, which introduces a tradeoff. In typical progressive 3-D mesh compression algorithms, the number of source coding bits can be reduced by either using *coarser quantizers* for the geometry coordinates or by sending *fewer polygons* (triangles in this case). The *choice* between these two approaches is also studied in this paper, and the proposed solution optimizes the decoded model quality by using the best combination of both methods. To summarize, this paper addresses the following problem:

Given: 1) the total bit budget (B bits) that consists of S source coding bits and C channel coding bits; 2) the channel packet-loss rate (P_{LR}); and 3) a 3-D model, find: 1) the number of triangles to send; 2) the step size of the geometry coordinates quantizer; 3) the total number of channel coding bits (C); and 4) the FEC codes to be assigned to every level (i.e., $C^{(j)}$), so that the expected distortion introduced in the decoded model is minimum.

We compute the above four quantities jointly to maximize the decoded model quality. First, we optimize the quantizer step size and the number of transmitted triangles together with the optimal (C) that gives the best expected quality. Then, we distribute C over the transmitted levels of the bitstream so that the more important parts get more error protection. This is done using a statistical distortion measure that is derived in Section III-C.

This paper is organized as follows. Section II describes the compression algorithm, the FEC codes, and the packetization method used in this paper. Section III presents the proposed joint source and channel coding method for UEP as well as the bit-allocation algorithm. Experimental results comparing no error protection (NEP), EEP, and UEP approaches are presented in Section IV. Finally, the conclusions are presented in Section V.

II. Background

A considerable amount of research has been recently invested in 3-D mesh compression. Deering [3] proposed a 3-D compression algorithm where triangle strips are traversed and the most recent traversed vertices are saved in a buffer. In this compression algorithm, normals and coordinates are predicted and quantized. In 1998, Taubin and Rossignac proposed the topological surgery [1] compression algorithm that has been adopted by MPEG-4 [16]. The idea of this method is to cut the model in a peeling-orange-skin style. First, the algorithm locates the edges to be cut. The result of cutting through these edges is a triangulated, simply connected polygon. Since then, several single-level compression algorithms have been proposed in the literature [2], [4], [11]. All of these single-level compression techniques do not account for the channel characteristics. Moreover, the received model is not displayed on the client's screen before the whole compressed bitstream is fully downloaded. This introduces a delay, specially when the model is transmitted over low-bandwidth channels. As a result, researchers investigated multilevel 3-D compression techniques to produce a bitstream that is scalable with respect to the channel bandwidth. There is a twofold approach in this direction. The first approach is to generate two bitstreams; one is a coarse representation of the model and the other is the full-resolution bitstream. The coarse bitstream is sent first and then the full-resolution bitstream is sent later. As a result, the client will have a coarse representation of the model to manipulate until the full-resolution bitstream is fully downloaded to replace the coarse model. Even though this method reduces the waiting time on the client side, it uses the bandwidth inefficiently because, once the full-resolution model is received, the coarse model is discarded.

The second approach is to generate a coarse representation and send it first to be displayed on the client's screen. Then, refinement information is streamed out to refine the coarse model until the full-resolution model is thoroughly displayed on the client's screen. Not only is the bitstream generated by this method scalable with respect to the channel bandwidth, but also the client can manipulate the coarse representation of the model while more refinements are downloaded and rendered as they are decoded. The latter method is more efficient than the former, and hence it has received attention from the research community.

Several progressive compression techniques have been proposed in the literature. The first algorithm is the progressive forest split compression algorithm [19]. Then, Kuo *et al.* [40] proposed a constructive traversal compression algorithm where a model is partitioned into segments. These segments are transmitted independently in addition to the boundary information. Such a technique has been adopted by MPEG-4 to provide a scalable bitstream [16]. Nevertheless, these partitions are encoded at full resolution.

Khodakovsky *et al.* [29] proposed a progressive compression technique that resamples the model to generate a regular mesh. To resample the model, the authors applied the multiresolution adaptive parameterization of surfaces (MAPS) algorithm [30] to move vertices tangentially to the model surface. Then, the geometry information is quantized and encoded using Loop

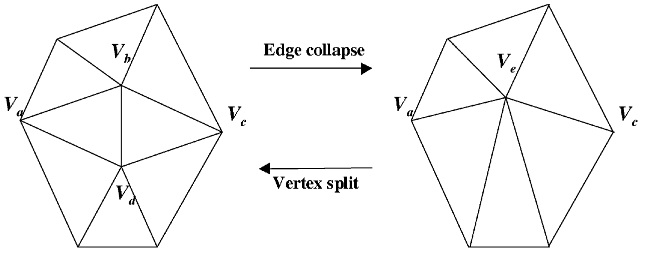


Fig. 1. *Edge-collapse* and *vertex-split* operations. Note that $V_e = (V_b + V_d)/2$.

wavelets and a zero-tree structure. Even though this method improves the compression ratio over other progressive methods, it requires the MAPS algorithm, which complicates the encoder substantially.

Finally, one of the simplest and most effective multilevel compression algorithms is the compressed progressive meshes (CPM) algorithm that was developed by Pajarola and Rossignac [20] in 1999. Because of its effectiveness, this method is used in this paper, although the proposed bit-allocation algorithm is also applicable to other progressive compression methods with minor modifications [21], [32]. This section summarizes the CPM algorithm to provide insight into the encoded bitstream content and the relative importance of its components. Also, the FEC codes that are used in this paper are briefly described later in this section. Finally, the packetization scheme employed in this paper is described.

A. Compressed Progressive Meshes

In this section, a brief description of the bitstream content produced by the CPM algorithm is presented [20]. The CPM algorithm depends on two operations: *edge-collapse* and *vertex-split* [32], [33], which are applied at the encoder and the decoder, respectively. These two operations are illustrated in Fig. 1. The CPM encoder and decoder block diagrams are depicted in Fig. 2.

The encoding process is iterative. Each iteration produces a coarser level of detail (LOD) of the model from a given level of detail (LOD). Also, each iteration produces a batch (i.e., an encoded bitstream) that is used by the decoder to produce the fine LOD from the coarser one. At the beginning of each iteration, a subset of edges is chosen to be collapsed. These edges have to satisfy certain conditions such that they can be collapsed within the current LOD. These conditions are as follows.

- At most two vertices are collapsed into one vertex.
- Let edge e_1 connect vertices v_1 and v_2, and similarly let edge e_2 connect vertices u_1 and u_2. If these four vertices form a quadrilateral, then these two edges cannot be collapsed within the same LOD.

These restrictions force the edges being collapsed to be independent of each other, and hence, the decoding process (*vertex-split* operation) for a given vertex is independent of others. Nevertheless, these restrictions limit the compression rate between consecutive LODs. Simulations show that a reduction of about 20%∼30% in the number of vertices between two consecutive LODs is achieved for a typical model.

Subject to the above restrictions, the errors that would be introduced on the model by collapsing each edge are updated after each batch, and the edges with the smallest errors are selected to be collapsed for the next batch. The error metric used here is computed as 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 of the vertices that collapsed to V [33], [34].

After generating the bitstream for the collapse operations of a certain LOD (i.e., a batch), the bitstream is appended to the front of the bitstream being assembled for the complete mesh. Within each batch, the bitstream is sorted according to the lexicographical order of the vertices in the coarser LOD. Both the encoder and the decoder should have the same ordering of the vertices at the beginning of each iteration [5], [35].

Each *edge-collapse* operation is represented by *three* fields. These three fields contain two types of information, namely *Geometry* and *Connectivity* information. These fields are as follows.

Collapse Status: One bit specifies whether a vertex is collapsed (1) or not (0). When this bit is (1), then the decoder continues decoding the next two fields; otherwise, it assumes that the two fields are empty and it proceeds to the next bit that corresponds to the collapse status of the next vertex.

Cut Edges: The decoder needs to decide which of the incident edges is to be split. These edges are called *cut edges*. For a vertex with e incident edges, $\left\lceil \log_2 \binom{e}{2} \right\rceil$ bits can specify any combination of two out of these e edges. These bits are appended to the bitstream. Note that the decoder knows the value of e for the current vertex.

Position Error: The difference in geometric coordinates between the collapsed vertex and the vertex locations estimated by the prediction algorithm. This difference between coordinates is quantized and entropy coded and the resulting three variable length codes (VLCs) are appended to the bitstream of the current batch.

The first two fields correspond to the *connectivity* information in the encoded bitstream while the third field is considered *geometry* information.

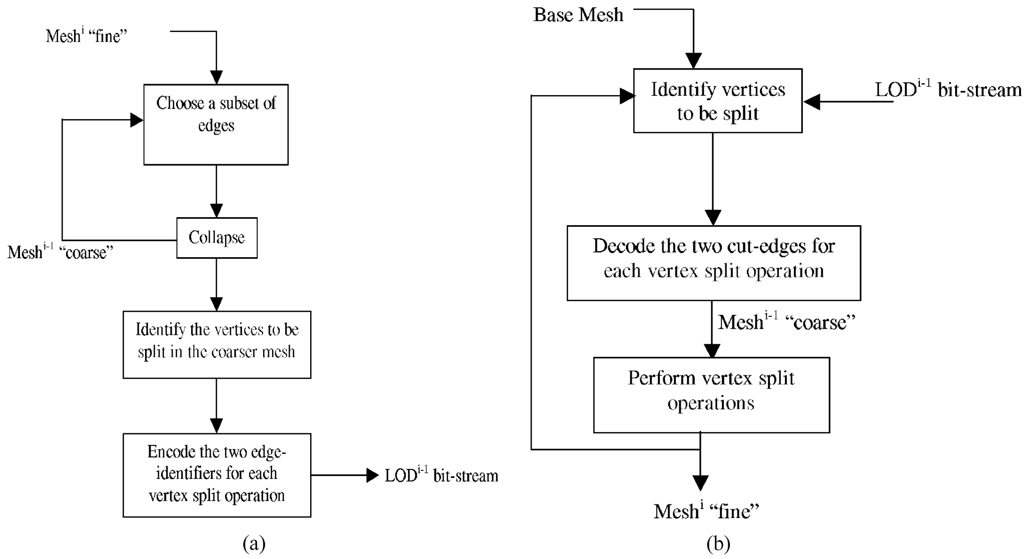
The CPM algorithm produces M batches in addition to the base mesh. The application of each batch yields an enhancing LOD that further approximates the original 3-D mesh. All edge-collapse operations are encoded using the above three fields. However, the base mesh can be compressed using any single-level mesh compression technique. In this paper, the base mesh is compressed using the TG algorithm [2].

At the decoder, the base mesh is decoded first. Then, the batches are decoded one at a time starting with the coarsest batch. While decoding each batch, a series of *vertex-split* operations is performed to add new vertices to the mesh.

Although we used the CPM algorithm to generate a scalable bitstream, our proposed method can be applied to any scalable bitstream generated by any progressive compression algorithm with minor modifications.

B. Reed-Solomon Codes

The FEC codes used in this paper are Reed–Solomon (RS) codes. These block codes are perfectly suited for error protection against bursty packet losses because they are maximum


Mesh^{i} "fine"
Choose a subset of edges
Collapse
Mesh^{i-1} "coarse"
Identify the vertices to be split in the coarser mesh
Encode the two edge-identifiers for each vertex split operation
LOD^{i-1} bit-stream
Base Mesh
Identify vertices to be split
LOD^{i-1} bit-stream
Decode the two cut-edges for each vertex split operation
Mesh^{i-1} "coarse"
Perform vertex split operations
Mesh^{i} "fine"

(a) (b)

Fig. 2. Block diagram of CPM encoder and decoder. (a) CPM encoder. (b) CPM decoder.

distance separable codes, i.e., there are no other codes that can reconstruct erased symbols from a smaller number of code symbols [36]. An RS code (n,k) defined over the Galois Field $GF(2^q)$ encodes k information symbols where each symbol contains q bits. These k symbols are combined with $n-k$ additional q-bit symbols into a codeword of n symbols. The codeword length n is restricted by $n \leq 2^q - 1$. In this paper, we chose q to be 8 b and therefore $n \leq 100$ (i.e., the RS codes used in here are defined over $GF(2^8)$). Moreover, we chose $n = 255$ symbols. The code rate of an (n,k) RS code is k/n, which represents the fraction to which the source data rate has to be reduced in order to maintain a constant total (source and channel) data rate. As soon as k symbols, out of each group of n, are received, all lost symbols can be reconstructed. Thus, if we expect to lose 5% of the packets, then we roughly choose $k/n = 0.95$ to be able to recover these lost packets. This number increases to 0.99 if we expect a packet-loss rate of 1%.

C. Packetization

In our study, we adapt a packetization method known as block of packets (BOP) [37]–[39]. In this method, the data is placed in horizontal streams and then FEC is applied on these streams vertically. The packets, however, are transmitted horizontally. Such a method is most appropriate for packet networks where burst errors are common [37], [38]. Fig. 3 is an example of a typical BOP where an (n,k) RS code is used to protect the transmitted bitstream.

Consider a scalable encoded bitstream that consists of four levels as shown in Fig. 4. The importance of the levels varies. The base mesh is the most important layer while batch 3 is the least important part of the encoded bitstream. Notice that, in this example, the size of the layers vary but we kept them fixed for illustration purposes. In order to reflect the variation in the levels' importance, we need to design the RS codes such that the RS code assigned for the base mesh (n,k_0) is more powerful than the RS codes assigned to the other layers. Similarly, the RS code assigned for the coarsest enhancement layer (n,k_1) is more powerful than the RS codes assigned to the finer enhancement layers, i.e., (n,k_2) and (n,k_3). We mathematically determine the RS code rates according to a statistical distortion measure that we derive in the following section.

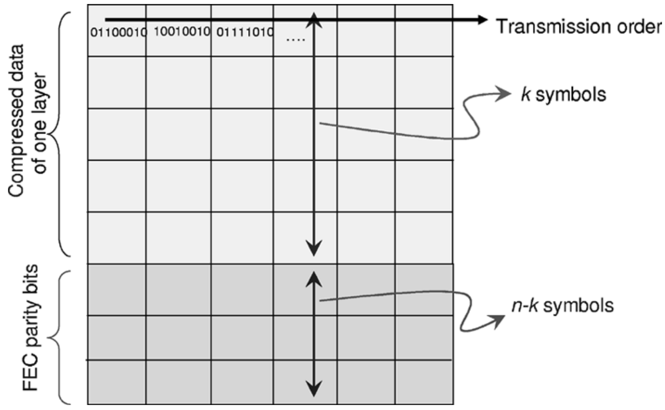


Fig. 3. Typical block of packets.

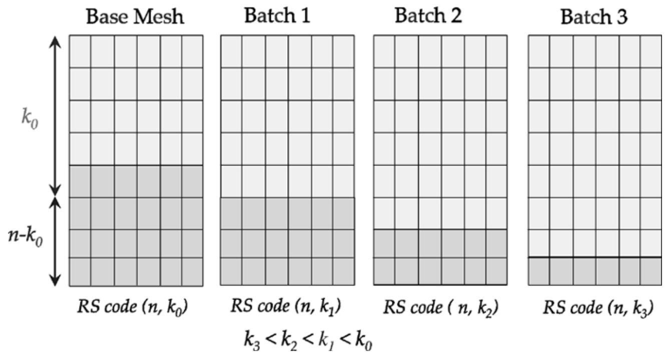


Fig. 4. Packetization example of a model encoded into a base mesh and three enhancement layers.

III. Proposed Method

The proposed UEP method is scalable with respect to both source and channel characteristics. Given: 1) an M-level progressively compressed 3-D model; 2) the total bit budget; and 3)

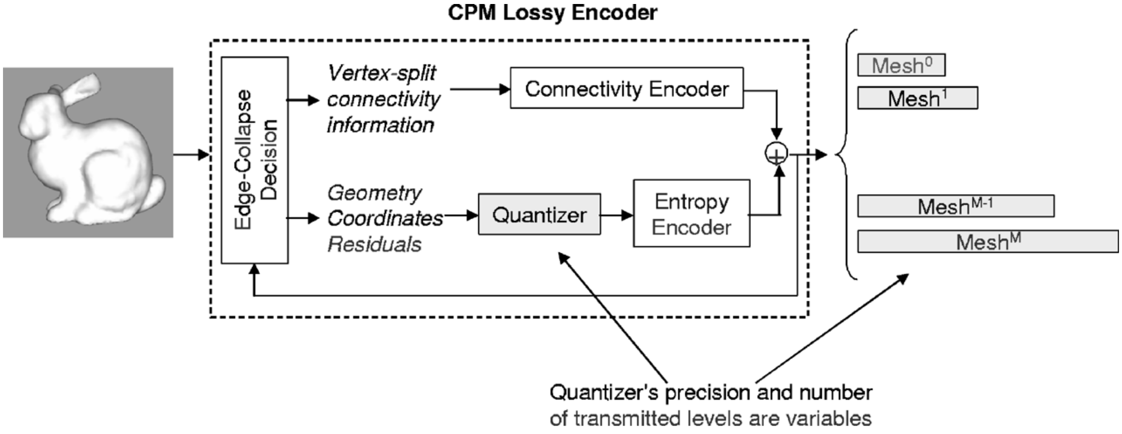

CPM Lossy Encoder
Edge-Collapse Decision
Vertex-split connectivity information
Connectivity Encoder
Geometry Coordinates Residuals
Quantizer
Entropy Encoder
Mesh^0
Mesh^1
Mesh^{M-1}
Mesh^M
Quantizer's precision and number of transmitted levels are variables

Fig. 5. Detailed block diagram of an M-level CPM encoder.

the channel packet-loss rate (P_{LR}), the proposed method calculates the optimal combination of: 1) the number of transmitted levels ($L \leq M$), which corresponds to the number of transmitted triangles; 2) the geometry coordinates quantizer step size (l-bit); 3) the total number of channel coding bits (C); and 4) the distribution of these channel coding bits over the transmitted levels (i.e., $C^{(j)}$, for $j = 0, \ldots, L$).[2] In order to achieve this, we need to allocate the bit budget to both codes through a joint source and channel coding method. In order to keep the total bandwidth unchanged, we need to reduce the number of source coding bits. As mentioned before, the number of source coding bits can be reduced either by using coarser geometry quantizers or by reducing the tessellation of the transmitted mesh. In this section, we first discuss the importance of tessellation versus geometry precision. Then, we show how we jointly optimize the above four parameters in order to maximize the expected decoded model quality. Afterward, we explain the statistical distortion measure that we developed to measure the expected distortion at the decoded model for a given packet-loss rate (P_{LR}). Finally, we present the solution of the optimization problem for a given bit budget (B).

A. Tessellation Versus Quantization

The detailed block diagram of the CPM encoder is depicted in Fig. 5. Notice that quantization is applied to the geometry residuals resulting from a prediction algorithm.

In order to keep the total bit budget unchanged when FEC is applied, the number of source coding bits needs to be reduced by a percentage amount equivalent to the applied FEC code rate. As shown in Fig. 5, the size of the encoded bitstream can be controlled by either changing the quantizer or reducing the number of transmitted batches (or, equivalently, the number of transmitted triangles). Although both operations control the size of the transmitted bitstream, they affect the decoded mesh quality in different ways. In order to measure the effect of each of these two size-reduction methods on the decoded mesh quality, we fixed the number of batches and used a spectrum of quantizers to encode the mesh. In each case, we decode all batches and calculate the maximum error between the mesh resulting from decoding each batch and the transmitted mesh. As an objective metric, we used the *Hausdorff* distance, which estimates the maximum distance between any two given sets of points E and F and is given by

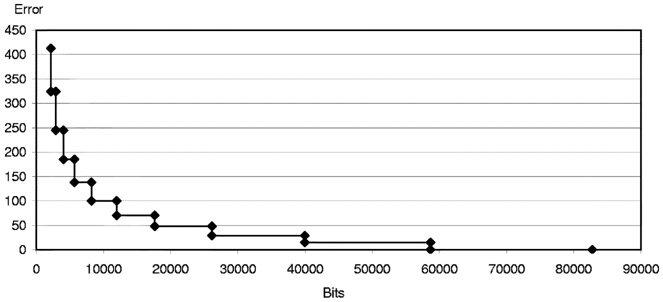


Fig. 6. Estimated error (using the *Hausdorff* distance) between the transmitted SMALL BUNNY model and the meshes decoded at different LODs. The model has been compressed into 10 batches and the base mesh (i.e., $M = 10$).

$$H(E, F) = \max(h(E, F), h(F, E)) \tag{1}$$

where

$$h(E, F) = \max_{e \in E} \min_{f \in F} \|e - f\|$$

and $\|\cdot\|$ denotes the Euclidean distance [40]. To get an accurate measure, we oversample all of the faces (i.e., triangles) of the two meshes to produce E and F before calculating the *Hausdorff* distance.

The resulting curve from these calculations is the rate-distortion (R-D) curve that estimates the distortion as a function of the number of received bits. Fig. 6 is a typical R-D curve that depicts the distortion at the decoded mesh as a function of the number of received bits. An R-D curve reflects the relative importance of the different parts of the encoded bitstream.

These R-D curves reflect two major facts. First, for a given bit budget to be spent on source coding only (i.e., $S = B$), a coarsely quantized bitstream with more triangles is better than a finer quantized bitstream with fewer triangles. For example, if the progressively compressed TRICERATOPS model is to be transmitted with a bit budget of 40 000 b, then, according to Fig. 7(a), we have the choice of sending: 1) nine batches of the 10-b quantized stream; 2) nine batches of the 8-b quantized stream; 3) 10

[2] We refer to the set of $C^{(j)}$'s as the vector $\mathbf{C_L}$.

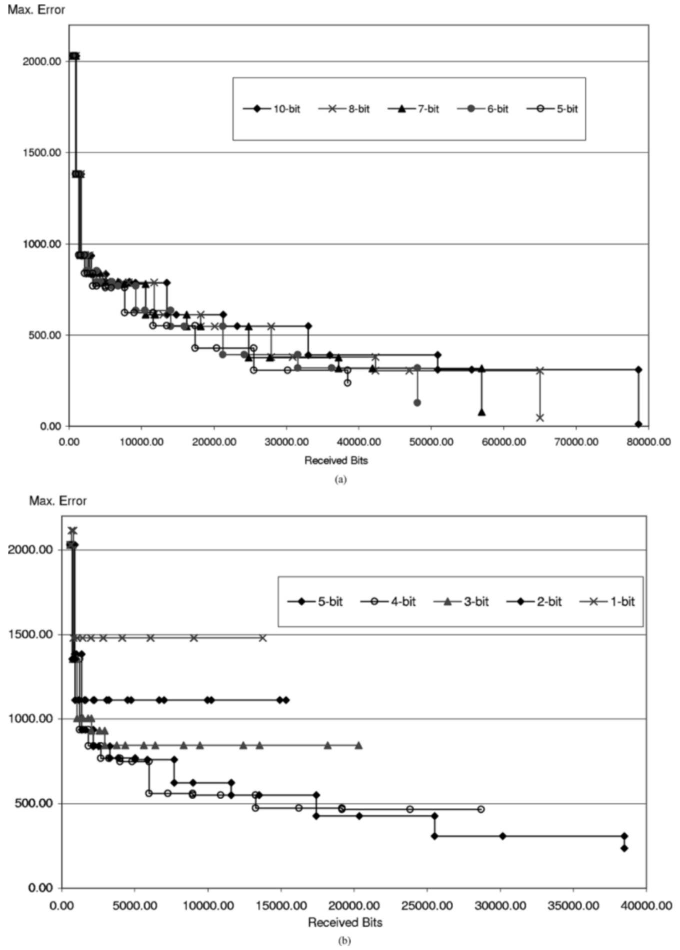


Fig. 7. R-D curves for the TRICERATOPS model that is compressed into 10 batches using different quantizers. The maximum error is measured using the *Hausdorff* distance between overly sampled meshes. (a) R-D curves for the quantizers 10, 8, 7, 6, and 5 b. (b) R-D curves for the quantizers 5, 4, 3, 2, and 1 b.

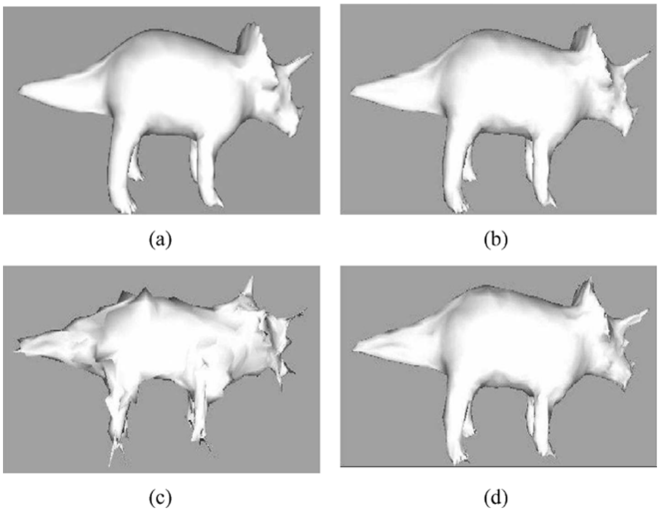


Fig. 8. Final decoded TRICERATOPS models for four different quantizers. (a) 8-b. (b) 5-b. (c) 2-b. (d) 4-b.

batches of the 7-b quantized stream; or 4) 10 batches of the 6-b quantized stream. The best choice among these is the one that gives the minimum distortion.[3] In this particular example, the best bitstream is the 6-b quantized one as shown in Fig. 7(a). Notice that in this discussion we study the effect of only source coding on the decoded model quality and, therefore, we assume that the channel is error-free. We incorporate the channel effects in the following section.

As another example, consider the case when the bit budget is 70 000 b. In this case, the best choice is to send 10 batches of the 10-b quantized bitstream. Therefore, based on the examples presented until now, we conclude that, for a given bit budget, the best encoded bitstream[4] is the coarsest-quantized one. In other words, using a coarser quantizer allows us to add more triangles than a finer quantizer for the same bit budget and hence tessellation is more important than the precision of the coordinates. This is true, however, only for a certain range of quantizers. As shown in Fig. 7(b), the performance of the 4-b quantizer is worse than that of the 5-b quantizer. Similarly, an $(l+1)$-bit quantizer is worse than an l-bit quantizer for l =1, 2, 3, 4 for the mesh under consideration. Thus, if the bit budget is 18 000 b, then the 5-b quantized bitstream performs better than all coarser and all finer quantized bitstreams in the *Hausdorff* distance sense. In other words, when a very coarse quantizer is used, tessellation will not improve the mesh quality. Thus, a second major conclusion from these R-D curves is that there exists a minimum quantization precision to be used to quantize the geometry coordinates residuals. In this paper, we determine this quantizer experimentally using R-D curves similar to the ones shown in Fig. 7.

In summary, given a 3-D model and a source coding bit budget, we need to follow the following steps in order to choose the best combination of geometry precision and number of transmitted batches.

- Determine the minimum quantizer after which tessellation does not improve the mesh quality.
- Determine all encoded bitstreams that can meet the given bit budget.
- Choose the coarsest quantized bitstream among these as the optimum bitstream.

Fig. 7 depicts the R-D curves for the TRICERATOPS model when l-bit quantizers (where $l \in \{10, 8, 7, 6, 5, 4, 3, 2, 1\}$) were used and the number of batches (M) is fixed to 10. For clarity, we split R-D curves into two plots.

These claims are supported by subjective measures as shown in Fig. 8. In these figures, the TRICERATOPS model has been progressively compressed into 10 batches, and each time a different quantizer is applied to the geometry coordinates residuals. In this figure, the final decoded model is shown when the 8-b, 5-b, 4-b, and 2-b quantizers are used. It can be seen from these figures that there is a significant drop in the decoded model quality when quantizers coarser than 5 b are used. Such a coarsest usable quantizer depends on the model under consideration; for most 3-D models we tested, this quantizer is found to have a step size between 4 and 6 b.

[3] Notice that the *Hausdorff* distance measures the maximum distance between two sets of points and, therefore, for two different encoding methods, the measured maximum distortion might be identical.

[4] The best in here is interpreted in terms of the *Hausdorff* distance.

So far, we have not included the channel coding bits into the bit budget. The next section combines FEC codes together with the R-D curves in order to get the best combination of the number of channel coding bits (C), geometry coordinates precision (l-bit quantizer), and the number of transmitted batches (L).

B. Joint Source and Channel Coding

Incorporating FEC codes within the analysis in Section III-A leads to the optimal solution we are after. For convenience, we restate the problem this paper addresses. Given a 3-D model and a total bit budget B,[5] determine an optimal combination of the following parameters to maximize the expected decoded model quality:

1) the geometry coordinates precision (i.e., what l needs to be for the l-bit quntizer used in quantizing the geometry coordinates prediction error);
2) the number of transmitted batches (i.e., L), which corresponds to the number of transmitted triangles;
3) the total number of channel coding bits (i.e., C);
4) the vector of FEC codes to be applied on the transmitted batches (i.e., $C_L = [C^{(0)}, C^{(1)}, \ldots, C^{(L)}]$). In other words, the distribution of C bits over the L transmitted batches.

In the previous section, we showed how to find the optimal combination of the first two parameters (i.e., l and L) for a given number of source coding bits (S). In order to incorporate C and $\mathbf{C_L}$ to solve the above problem, we run a search algorithm that finds the best combination of the first three parameters (l, L, C) as well as a local search algorithm that determines the fourth parameter ($\mathbf{C_L}$). At every step, we increase the number of channel coding bits by a predetermined increment and we find the best pair (l, L) from R-D curves for the corresponding number of source coding bits (S). Then, using (C, l, L) together with the packet-loss rate (P_{LR}), we search for the best distribution of the C bits over the transmitted L batches.

For clarity, the main algorithm is highlighted in Fig. 9. As shown in the figure, we used a distortion measure that estimates the distortion introduced at the decoded model for a given l-bit quantizer, L batches, channel coding bits (C) and the channel packet-loss rate (P_{LR}). The result of optimizing this distortion measure with respect to $\mathbf{C_L}$ and P_{LR} is an optimum set of RS code rates to be applied on the transmitted levels, i.e., $\mathbf{C_{L_{optimum}}}$. The following two sections discuss this distortion measure followed by the solution of the bit-allocation algorithm for typical 3-D models.

C. Statistical Distortion Measure

All UEP methods regardless of their application area rely on a distortion measure that quantifies the relative importance of bits. Therefore, the success of our method naturally depends on how accurately and effectively the proposed measure characterizes the distortion that is introduced when certain bits are lost. In general, mesh distortion estimation depends on the decoding strategy. Because of the sensitivity of connectivity information within batches, the decoding process terminates whenever a connectivity packet is lost, and therefore all succeeding packets are discarded.

```
INPUT:  total bit budget ($B$ bits),
        channel packet-loss rate ($P_{LR}$),
        a set of $l$-bit quantizers, and
        the R-D curves for these quantizers.

Let $Q$ be a variable that represents the step size of the global search algorithm.

- Start with S = B and C = 0
        From R-D curves, find the best pair $(l, L)$ for this $S$.
        Compute $(D^{(0)}, \mathbf{C_L^{(0)}}) = \text{Min } D_r(l, L, C, P_{LR})$ using a local search algorithm.
- Set S = B-Q and C = Q
        From R-D curves, find the best pair $(l, L)$ for this $S$.
        Compute $(D^{(1)}, \mathbf{C_L^{(1)}}) = \text{Min } D_r(l, L, C, P_{LR})$ using a local search algorithm.
- Set S = B-2Q and C = 2Q
        From R-D curves, find the best pair $(l, L)$ for this $S$.
        Compute $(D^{(2)}, \mathbf{C_L^{(2)}}) = \text{Min } D_r(l, L, C, P_{LR})$ using a local search algorithm.
        ⋮
        ⋮
- Set S = $S^{(0)}$ and C = B-$S^{(0)}$, where $S^{(0)}$ is the size of the base mesh in bits
        From R-D curves, find the best pair $(l, L)$ for this $S$.
        Compute $(D^{(\frac{B+S^{(0)}}{Q})}, , \mathbf{C_L^{(\frac{B+S^{(0)}}{Q})}}) = \text{Min } D_r(l, L, C, P_{LR})$ using a local search
algorithm.

OUTPUT: min($D^{(0)}$, $D^{(1)}$, ..., $D^{(\frac{B+S^{(0)}}{Q})}$) and the corresponding ($l$, $L$, $C$, and $\mathbf{C_L}$).
```

Fig. 9. Search algorithm for finding the best combination of l, L, C, and $\mathbf{C_L}$. $\mathbf{C_L} = [C^{(0)}, \ldots, C^{(L)}]$, where $C^{(j)}$ is the number of channel coding bits dedicated to the jth layer of the encoded bitstream.

Referring to Fig. 9, the developed distortion measure (D_r) estimates the distortion at the decoded mesh for a given packet-loss rate (P_{LR}) when an l-bit quantizer is used to quantize the geometry coordinates residuals, L batches are sent, and a total of C bits are dedicated for error protection. The local minimization algorithm searches for the best[6] distribution of the C bits over the transmitted $L + 1$ levels.

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 stops at *level-j*, and P_j is the probability of having an irrecoverable packet loss at *level-j*. We compute E_j from the R-D curves. For simplicity, we associate connectivity and geometry to the same distortion level even though geometry is slightly less important than connectivity since losing geometry does not necessarily terminate the decoding process.

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 *level-j*. Mathematically, the expected mesh distortion at the decoder can be written as shown in (2), the bottom of the next page, where L is the total number of transmitted levels, which is limited to $0 \leq L \leq M$, $M + 1$ is the number of encoded levels including the base mesh, P_j is the probability of having irrecoverable packets in the jth level, and E_j is the error between the transmitted mesh and the mesh produced by decoding the bitstream of *level-$(j - 1)$*.

In (2), P_j can also be defined as the probability of losing more than $n - k_j$ packets in the jth layer, since the RS code (n, k_j)

[5] B consists of S bits for source coding and C bits for channel coding.

[6] "Best" here is understood in terms of the derived distortion measure calculation.

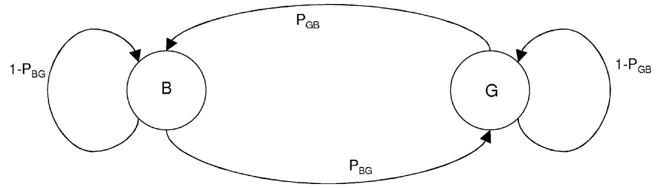


Fig. 10. G-E two-state Markovian channel model.

applied at this layer will recover from a fewer number of packet losses. This quantity can be calculated in terms of the block error density function as

$$P_j = \sum_{m=n-k_j+1}^{n} p(m,n) \tag{3}$$

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

So far, the only quantity in (2) that has not yet been calculated yet is the block error density function $p(m,n)$, which depends on the channel model used. This computation is detailed in the following section.

D. Determination of the Block Error Density Function $p(m,n)$

The bit error rate (BER) in (2) depends on the channel model under consideration. The underlying processes that lead to packet losses in networks are quite complex. However, even a simple and analytically tractable Markov model with only two states approximates the network packet loss behavior fairly well for our purpose. This model is illustrated in Fig. 10. The two-state Markovian Gilbert–Elliot (G-E) model has been thoroughly investigated in the literature [38], [37]. Here, we briefly restate the main results. Of particular interest within the scope of this paper is the probability of having m errors in n symbols when an RS code defined over $GF(2^q)$ with rate k/n is used. The G-E model possesses a characteristic distribution of error-free intervals, which are also called *gaps*. To be precise, the gap is defined as the interval of length $v-1$ packets between two consecutive erroneously received packets. Let $g(v)$ be the probability of having a gap of v received packets between two lost packets. Then, $g(v)$ is given by

$$g(v) = \begin{cases} 1 - p_{\mathrm{BG}}, & v = 1 \\ p_{\mathrm{BG}}(1-p_{\mathrm{GB}})^{v-2}p_{\mathrm{GB}}, & v > 1 \end{cases} \tag{4}$$

where p_{BG} represents the transition probability from the bad state to the good state while p_{GB} represents the transition probability from the good state to the bad state.

Similarly, let $G(v)$ be the probability of having a gap greater than $v-1$ packets. Then, $G(v)$ is given by

$$G(v) = \begin{cases} 1, & v = 1 \\ p_{\mathrm{BG}}(1-p_{\mathrm{GB}})^{v-2}, & v > 1 \end{cases}. \tag{5}$$

Furthermore, let $R(m,n)$ denote the probability of having $m-1$ packet losses within the $n-1$ packets following a lost packet. Then, $R(m,n)$ is given by

$$R(m,n) = \begin{cases} G(n), & m = 1 \\ \sum\limits_{v=1}^{n-m+1} g(v)R(m-1,n-v), & 2 \le m \le n \end{cases}. \tag{6}$$

Finally, the probability of losing m packets within a block of n packets can be written as

$$p(m,n) = \begin{cases} \sum\limits_{v=1}^{n-m+1} P_{\mathrm{LR}}G(v)R(m,n-v+1), & 1 \le m \le n \\ 1 - \sum\limits_{v=1}^{n} p(v,n), & m = 0 \end{cases} \tag{7}$$

where P_{LR} is the packet-loss rate.

Now, after estimating the quantities in (2), we move to solve the main bit-allocation algorithm highlighted in Section III-B to maximize the decoded model quality. This is detailed in the following section.

E. Solving for an Optimal Bit Allocation

In this section, we solve the local optimization problem that distributes C bits over the transmitted $L+1$ levels. Then, we solve the global optimization problem, as illustrated in Fig. 9, to determine (l, L, C, and $\mathbf{C_L}$).

Equation (2) estimates the expected distortion introduced at the decoded mesh in a statistical sense. Now, the objective is to optimally distribute C bits among the transmitted $L+1$ levels (i.e., $C^{(0)}, C^{(1)}, \ldots, C^{(L)}$) in order to minimize the distortion at the decoded mesh. Intuitively, the base mesh is usually regarded as the most important layer in the encoded bitstream, followed by the coarsest level, and so on, up to the finest level. Therefore, we expect 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 (2) and (3) must satisfy two conditions. First, the error protection bit budget is upper-bounded by C, the maximum number of available error protection bits. Second, k_j cannot be greater than n. Combining (2) together with these two conditions results in a constrained optimization problem given as follows:

$$\min \underset{l,L,C,C^{(j)}}{\arg} \mathcal{D}_r(L),$$

$$D_r(L) = \begin{cases} P_0E_0, & L = 1 \\ P_0E_0 + \sum\limits_{j=1}^{L-1} P_jE_j \prod\limits_{i=0}^{j-1}(1-P_i) + \prod\limits_{j=0}^{L-1}(1-P_j)E_L, & 1 < L < M \\ P_0E_0 + \sum\limits_{j=1}^{L-1} P_jE_j \prod\limits_{i=0}^{j-1}(1-P_i), & L = M \end{cases} \tag{2}$$

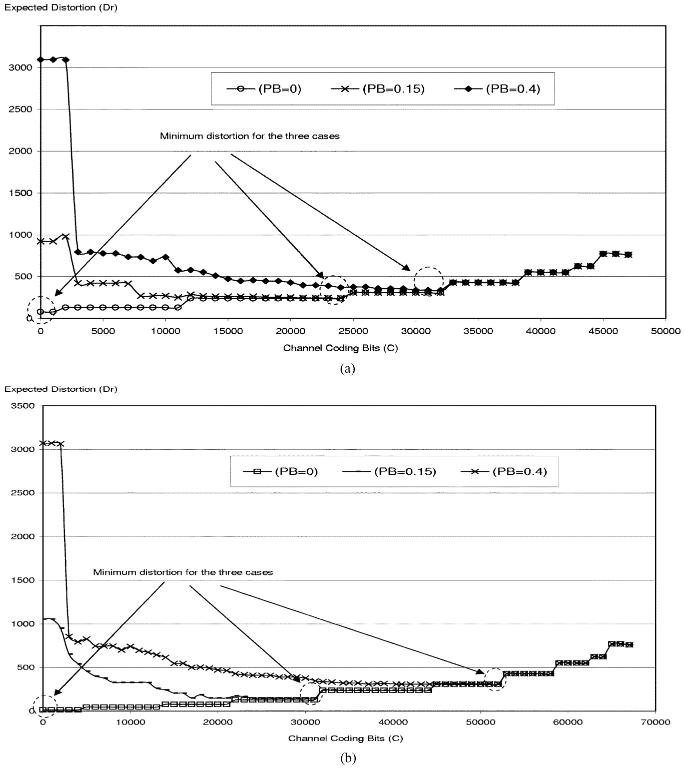


Fig. 11. Expected distortion as a function of the channel coding bits for two different total bit budgets and three packet-loss rates for the TRICERATOPS model. (a) Total bit budget (B) is 70 000 bits. (b) Total bit budget (B) is 50 000 bits.

subject to

$$\sum_{j=0}^{L-1} C^{(j)} = C \text{ and } 0 \leq k_j \leq n,$$

where

$$j = 0, \ldots, L-1. \qquad (8)$$

Replacing the local search algorithm in Fig. 9 with (2) and (8) completes the setup for the search algorithm highlighted in Section III-B. Now, we solve the global optimization algorithm to find the optimum combination of (l, L, C, and $\mathbf{C_L}$).

As an example, we solved the proposed optimum bit allocation algorithm for the TRICERATOPS model for two different bit budgets and a spectrum of packet-loss rates. In Fig. 11(a), we plot the estimated distortion (D_r) as a function of the number of channel coding bits (C) for three packet-loss rates (P_{LR}). In these plots, the step size Q (see the algorithm in Fig. 9) is chosen to be 1000 b. The total bit budget (B) in this case is set to 70 000 b. As shown in this plot, when the channel is error-free (i.e., $P_{\text{LR}} = 0\%$), the optimization algorithm dedicates zero bits to channel coding. However, when the packet-loss rate increases to 15%, the optimal number of channel coding bits is computed to be 31 000 b. Similarly, when the channel is subject to severe packet losses ($P_{\text{LR}} = 40\%$), C is computed to be 53 000. Another noticeable feature of the curves in Fig. 11(a) is that after a certain channel coding bit budget (>50 000 b) all three curves converge. At such high channel coding bits (>70% of the total bit budget), only the coarse batches of the 5-b quantized bit stream are transmitted and there are enough channel coding bits for error protection at the three packet-loss rates.

TABLE I
OPTIMIZATION SOLUTION FOR A SPECTRUM OF PACKET-LOSS RATES AND TWO TOTAL BIT BUDGETS FOR THE TRICERATOPS MODEL.

(a) Total bit budget = 70, 000.

P_{LR}	$C_{optimal}$	$L_{optimal}$	Opt. Quantizer	$D_{r_{min}}$
0.00	0	10	10-bit	13.38
0.08	21000	10	7-bit	94.83
0.15	31000	10	6-bit	147.30
0.40	53000	10	5-bit	214.136

(b) Total bit budget = 50, 000.

P_{LR}	$C_{optimal}$	$L_{optimal}$	Opt. Quantizer	$D_{r_{min}}$
0.00	0	10	7-bit	78.16
0.08	11000	10	6-bit	164.73
0.15	24000	10	5-bit	243.362
0.40	31000	9	5-bit	291.295

The optimum number of channel coding bits (C_{optimal}) together with the optimal quantizer step size (l-bit), the optimal number batches to be sent (L_{optimal}), and the resulting minimum distortion ($D_{r_{\text{optimal}}}$) are listed in Table I(a) for the TRICERATOPS model when the total bit budget (B) is set to 70 000 b. As expected, at the high packet-loss rate, the coarsest-quantized bitstream is used. When the packet-loss rate increases, more channel coding bits will be assigned and hence fewer batches will be sent from the 5-b quantized bitstream.

The same optimization problem has been solved for the TRICERATOPS model when the total bit budget (B) is 50 000 b. The resulting plot and table are shown in Fig. 11(b) and Table I(b), respectively.

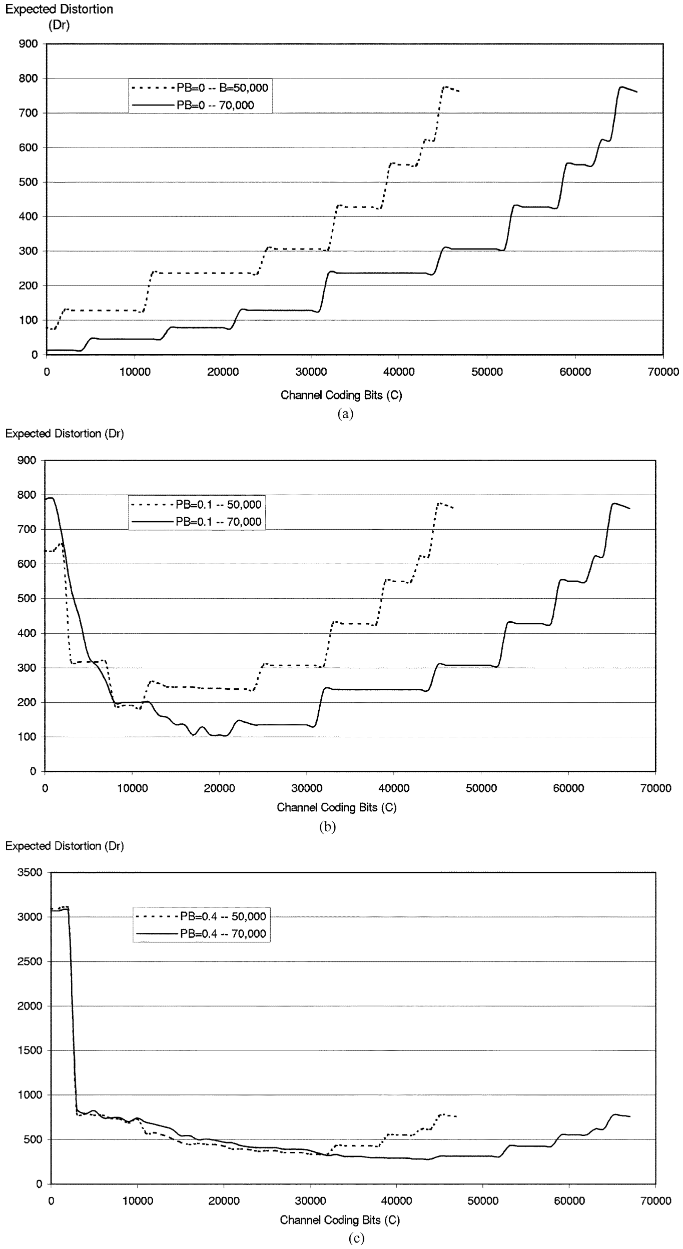
Another interesting observation of the bit-allocation solution is the difference between the estimated distortions for the same packet-loss rate but for different bit budgets. This difference is depicted in Fig. 12 for three packet-loss rates. When the channel is error-free, the difference between the two estimated distortions for the two bit budgets (50 000 and 70 000) is large. This difference shrinks as the packet-loss rate increases as shown in Fig. 12(b) and (c), respectively, especially at low channel coding bits. This is anticipated with a low number of channel coding bits and a high packet-loss rate, the transmitted data is subject to high losses. Hence, the accuracy of the source coding bits does not improve the decoded model quality.

Finally, Fig. 13 depicts the computed number of optimal channel coding bits (i.e., C_{optimal}) as a function of the packet-loss rate (P_{LR}) for the cases when the total bit-budget (B) is 70 000 and 50 000 b, respectively. As shown at any packet-loss rate, the channel coding bit budget of the former case is higher than the latter. This is expected since in the former case, more bits are available for both error protection and source coding accuracy.

Now, the optimum parameters from the bit-allocation algorithm are used to send the encoded bitstream. In the following section, we run simulations to compare our proposed UEP method that uses the proposed bit-allocation algorithm with the methods of EEP and NEP.

IV. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of the proposed UEP algorithm, we used both subjective and objective methods of com-



(a)

(b)

(c)

Fig. 12. Expected distortion as a function of the number of channel coding bits for two different total bit budgets and three packet-loss rates for the TRICERATOPS model. (a) $P_{\mathrm{LR}} = 0.00$. (b) $P_{\mathrm{LR}} = 0.10$. (c) $P_{\mathrm{LR}} = 0.40$.

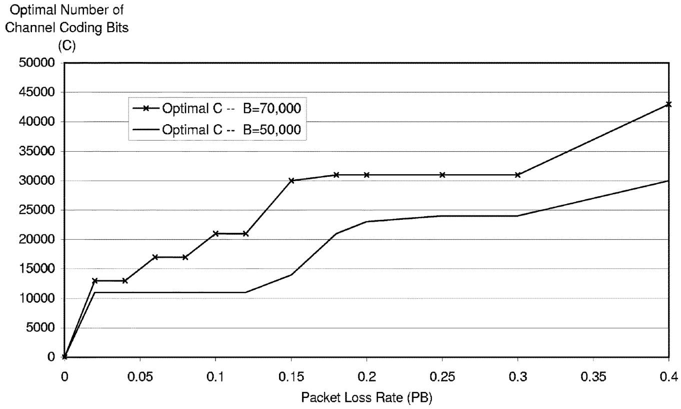
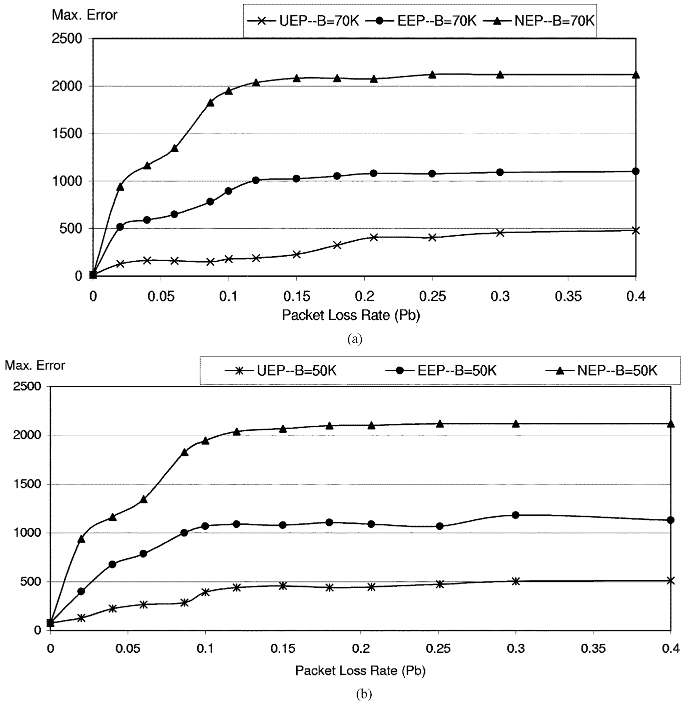


Fig. 13. The computed optimal channel coding bit budget (C_{optimal}) for a spectrum of packet-loss rates for the TRICERATOPS model. Two total bit budgets are used: 70 000 and 50 000 b.



(a)

(b)

Fig. 14. Simulation results when UEP, EEP, and NEP methods are applied on the TRICERATOPS model for two total bit budgets. ($L_B = 5.0$). (a) Total bit budget (B) is 70 000 b. (b) Total bit budget (B) is 50 000 b.

parison. In particular, we used the *Hausdorff* distance 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. For fair comparison between the three methods (i.e., NEP, EEP, and UEP), the total number of bits is kept the same.

We applied the proposed UEP method on the TRICERATOPS model. First we compressed it progressively using the CPM algorithm. In all experiments discussed in this section, the average burst length, L_B, is set to 5 and the RS packet size is set to 100 (i.e., $n = 100$).

Fig. 14(a) depicts the maximum error between the transmitted and the decoded meshes as a function of the packet-loss rate (P_{LR}) for the TRICERATOPS model. Three curves are shown for the UEP, EEP, and NEP methods. The total bit budget is chosen to be 70 000 b. As can be seen from these curves, when the channel is error-free, all three methods experience the same distortion, which results primarily from quantization. However, when the packet-loss rate increases, the proposed UEP method shows a more graceful degradation compared to the other two methods. An important observation from these curves is that after a packet-loss rate of 20%, the experienced degradation for any method does not change considerably. This is because at such high packet-loss rates a major part of the connectivity as well as geometry is lost and hence the decoded mesh is far from the transmitted one in the *Hausdorff* distance sense. The parameters used in the UEP case are tabulated in Table I(a) for a set of packet-loss rates. The corresponding subjective results are also shown in Fig. 16. These results show that the proposed UEP

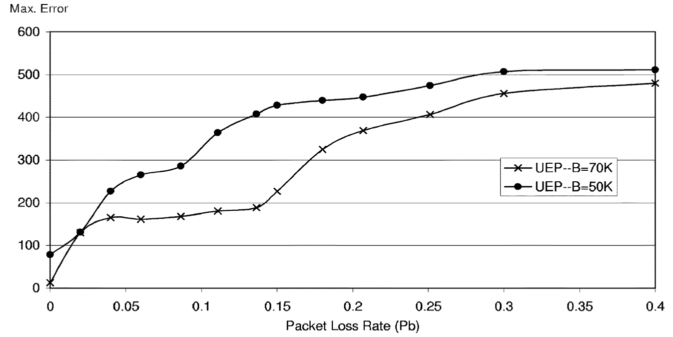


Fig. 15. Simulation results when UEP methods is applied on the TRICERATOPS model for two total bit budgets: 70 000 and 50 000 b. ($L_B = 5.0$).

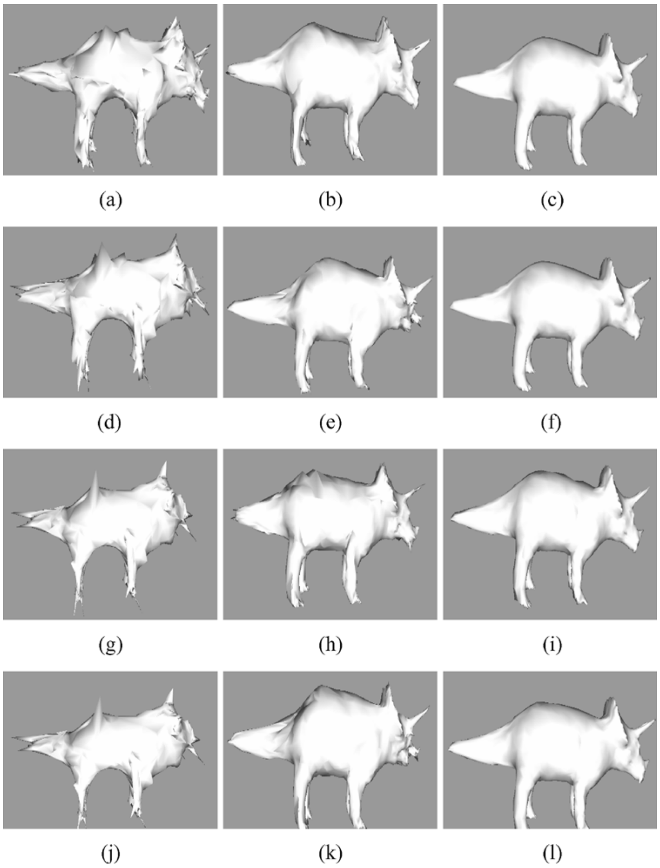

(a) (b) (c)
(d) (e) (f)
(g) (h) (i)
(j) (k) (l)

Fig. 16. Subjective results of applying NEP, EEP, and UEP methods on the TRICERATOPS model where the total bit budget is 70 000 b. (a) NEP $P_{\mathrm{LR}} = 0.06$. (b) EEP $P_{\mathrm{LR}} = 0.06$. (c) UEP $P_{\mathrm{LR}} = 0.06$. (d) NEP - $P_{\mathrm{LR}} = 0.10$. (e) EEP - $P_{\mathrm{LR}} = 0.10$. (f) UEP $P_{\mathrm{LR}} = 0.10$. (g) NEP $P_{\mathrm{LR}} = 0.20$. (h) EEP $P_{\mathrm{LR}} = 0.20$. (i) UEP $P_{\mathrm{LR}} = 0.20$. (j) NEP $P_{\mathrm{LR}} = 0.40$. (k) EEP $P_{\mathrm{LR}} = 0.40$. (l) UEP $P_{\mathrm{LR}} = 0.40$.

method kept a considerable quality level of the decoded mesh compared to both EEP and NEP.

The same experiments have been repeated on the same mesh but with different total bit budgets. In this case, the total bit budget was chosen to be 50 000 b. The resulting maximum error curves for the three methods, UEP, EEP and NEP, are depicted in Fig. 14(b). The corresponding subjective results are shown in Fig. 17. The trend in these results is similar to the previous case that used a total bit budget of 70 000 b. However, there is a difference in the distortion levels between the curves in Fig. 14(a) and the ones in Fig. 14(b). Fig. 15 depicts this difference for the UEP method. As shown, when the channel is error-free, the UEP case with a total bit budget of 70 000 b performs better than when the total bit budget is 50 000 b. This is expected since no channel coding bits are used and hence all bits are utilized by source coding. Thereafter, the former case uses more bits, which corresponds to finer quantizers and results in less distortion at the decoded mesh. As the channel packet-loss rate increases, the UEP case with a total bit budget of 70 000 b continues to outperform the case when the total bit budget is 50 000 b. Nevertheless, when the packet-loss rate increases considerably (i.e., larger than 40%), the two cases show very similar performance. This is because, at such a high loss rate, all transmitted levels are subject to high loss and increasing the bit budget by 20 000 b does not improve the decoded mesh quality.

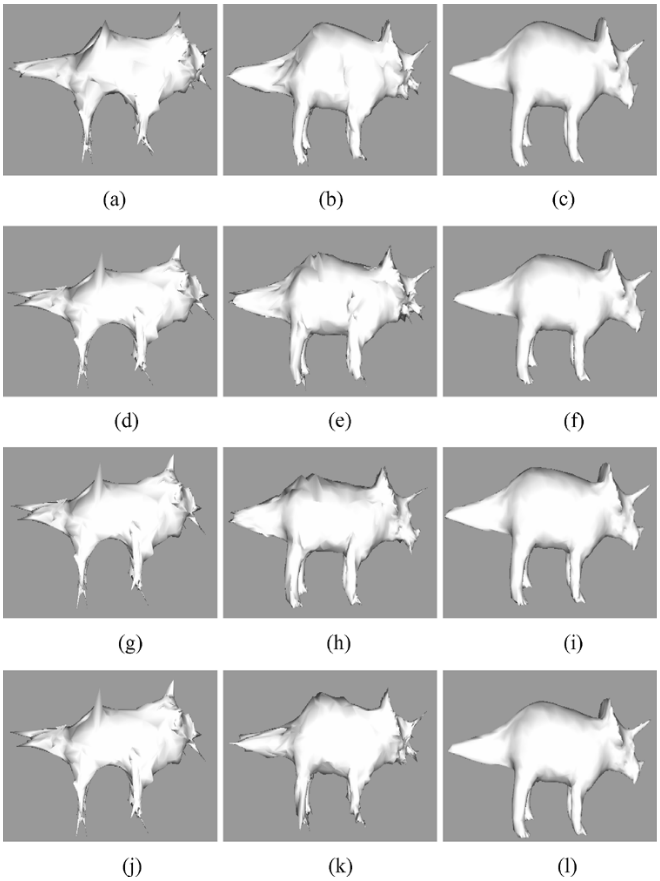

(a) (b) (c)
(d) (e) (f)
(g) (h) (i)
(j) (k) (l)

Fig. 17. Subjective results of applying NEP, EEP, and UEP methods on the TRICERATOPS model where the total bit budget is 50 000 b. (a) NEP $P_{\mathrm{LR}} = 0.06$. (b) EEP $P_{\mathrm{LR}} = 0.06$. (c) UEP $P_{\mathrm{LR}} = 0.06$. (d) NEP $P_{\mathrm{LR}} = 0.10$. (e) EEP $P_{\mathrm{LR}} = 0.10$. (f) UEP $P_{\mathrm{LR}} = 0.10$. (g) NEP $P_{\mathrm{LR}} = 0.20$. (h) EEP $P_{\mathrm{LR}} = 0.20$. (i) UEP $P_{\mathrm{LR}} = 0.20$. (j) NEP $P_{\mathrm{LR}} = 0.40$ (k) EEP $P_{LR} = 0.40$. (l) UEP $P_{\mathrm{LR}} = 0.40$.

V. Conclusion

In this paper, we proposed a bit-allocation algorithm that provides optimal source and channel code rates to maximize the expected decoded model quality. We used this algorithm in our proposed UEP method. The problem that is addressed can be stated as follows. Given a 3-D model and a total bit budget B, determine an optimal combination of the following parameters to maximize the decoded mesh quality:

1) The precision of the geometry coordinates (i.e., l).
2) The number of transmitted batches (i.e., L), which also defines to the number of transmitted triangles.
3) The total number of channel coding bits (i.e., C).
4) The vector of FEC codes to be applied on the transmitted batch (i.e., $C_L = [C^{(0)}, C^{(1)}, \ldots, C^{(L)}]$).

In order to quantify the distortion introduced at the decoded mesh for any given set of the aforementioned parameters, we developed a statistical distortion measure. Then, we optimize this distortion measure to estimate the optimal combination of these parameters. The resulting UEP solution provides a graceful degradation in the decoded mesh quality as the packet-loss rate increases. In the following, we will comment on a few features of the proposed 3-D mesh transmission system.

- 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. It is applicable wherever the proposed statistical distortion measure is calculable. Thus, the proposed framework can be used with other progressive 3-D mesh compression methods as well.
- The applicability of the proposed UEP method does not depend on a particular channel model, although we used the G-E model in this paper. The block error density function can be calculated for any channel model.
- The proposed 3-D mesh transmission system is more complicated than a standard system with no error protection since, for every 3-D model to be transmitted, we have to run the optimization algorithm highlighted in Fig. 9. However, this increase in complexity can be justified (for some applications) by the increase in the decoded mesh quality as illustrated by the experimental results. Furthermore, these preprocessing operations can be performed off-line, tabulated in look-up tables, and stored along with the 3-D graphics data. Also, in this paper, we determined the relation between quantization and tessellation using the R-D curves. The authors are currently working on developing a more efficient ways of producing these R-D curves.

References

[1] G. Taubin and J. Rossignac, “Geometric compression through topological surgery,” *ACM Trans. Graph.*, vol. 17, no. 2, pp. 84–115, Apr. 1998.

[2] C. Touma and C. Gotsman, “Triangle mesh compression,” in *Proc. Graphics Interface*, Vancouver, BC, Canada, Jun. 1998, pp. 26–34.

[3] M. Deering, “Geometry compression,” in *PROC. SIGGRAPH’95*, Aug. 1995, pp. 13–20.

[4] S. Gumhold and W. Strasser, “Real time compression of triangle mesh connectivity,” in *Proc. SIGGRAPH’98*, Jul. 1998, pp. 133–140.

[5] J. Rossignac, “Edgebreaker: Connectivity compression for triangle meshes,” *IEEE Trans. Vis. Comput. Graphics*, vol. 5, pp. 47–61, Jan. 1999.

[6] J. Rossignac and A. Szymczak, “Wrap&zip decompression of the connectivity of triangle meshes compressed with edgebreaker,” *J. Computat. Geometry, Theory Applicati.*, vol. 14, no. 1–3, pp. 119–135, Nov. 1999.

[7] J. Rossignac, A. Safanova, and A. Szymczak, “3-D compression made simple: Edgebreaker on a corner table,” in *Proc. Shape Modeling Int. Conf.*, Genoa, Italy, May 2001, pp. 278–283.

[8] A. Szymczak, D. King, and J. Rossignac, “An edgebreaker-based efficient compression scheme for connectivity of regular meshes,” *J. Computat. Geometry, Theory Applicati.*, vol. 20, no. 2, pp. 257–269, Oct. 2001.

[9] D. King and J. Rossignac, “Guaranteed 3.67 V bit encoding of planar triangle graphs,” in *Proc. 11th Canadian Conf. Computat. Geometry*, Vancouver, BC, Canada, Aug. 1999, pp. 146–149.

[10] J. Rossignac, A. Safanova, and A. Szymczak, “3-D compression made simple: Edgebreaker on a corner table,” in *Proc. Shape Modeling Int. Conf.*, Genoa, Italy, May 2001, pp. 278–283.

[11] M. Chow, “Optimized geometry compression for real-time rendering,” in *Proc. IEEE Visualization ’97*, Phoenix, AZ, Oct. 1997, pp. 347–354.

[12] G. Taubin, W. Horn, F. Lazarus, and J. Rossignac, “Geometry coding and vrml,” *Proc. IEEE*, vol. 96, no. 6, pp. 1228–1243, Jun. 1998.

[13] P. Doenges, T. K. Capin, F. Lavagetto, J. Ostermann, I. S. Pandzic, and E. D. Petajan, “MPEG-4: Audio/video and synthetic graphics/aduio for mixed media,” *Signal Process.: Image Commun.*, vol. 9, pp. 433–463, 1997.

[14] F. Pereira, “MPEG-4: Why, what, how and when?,” *Signal Process.: Image Commun.*, vol. 15, pp. 271–279, 2000.

[15] “SNHC Verification Model 9.0, “[3D Mesh Encoding]”,”, ISO/IEC JTC1/SC29/WG11 N2301, 1998.

[16] A. E. Walsh and M. B. Sevenier, *MPEG-4: Jump-Start*. Upper Saddle River, NJ: Prentice-Hall, 2002.

[17] J. Rossignac, “3-D compression,” in *Lecture at the ACM SIGGRAPH’99*, 1999.

[18] ——, “Shape complexity and compression,” in *Proc. Eurographics’00*, 2000, p. [AUTHOR: PLEASE PROVIDE PAGE NUMBERS.—ED.].

[19] G. Taubin, A. Gueziec, W. Horn, and F. Lazarus, “Progressive forest split compression,” in *Proc. SIGGRAPH’98*, Jul. 1998, pp. 123–132.

[20] R. Pajarola and J. Rossignac, “Compressed progressive meshes,” *IEEE Trans. Vis. Comput. Graphics*, vol. 6, no. 1–3, pp. 79–93, Jan.-Mar. 2000.

[21] J. Popovic and H. Hoppe, “Progressive simplicial complexes,” in *Proc. SIGGRAPH’97*, pp. 217–224.

[22] R. Pajarola and J. Rossignac, “Squeeze: Fast and progressive decompression of triangle meshes,” in *Proc. Int. Conf. Comput. Graphics*, Geneva, Switzerland, June 2000, pp. 173–182.

[23] H. Hoppe, “Efficient Implementation of Progressive Meshes,”, Microsoft Research Tech. Rep. (MSR-TR-98-2), 1998.

[24] G. Al-Regib, Y. Altunbasak, and J. Rossignac, “A joint source and channel coding approach for progressively compressed 3-D model transmission,” J. Graphics Models, September 2001, submitted for publication.

[25] Y. Wang and Q.-F. Zhu, “Error control and concealment for video communications: A review,” *Proc. IEEE*, vol. , no. 5, pp. 974–997, May 1998.

[26] G. Al-Regib and Y. Altunbasak, “An unequal error protection method for packet loss resilient 3-D mesh transmission,” in *Proc. IEEE INFOCOM*, vol. 2, New York, Jun. 2002, pp. 743–752.

[27] Z. Yan, S. Kumar, and C.-C. J. Kuo, “Error resilient coding of 3-D graphic models via adaptive mesh segmentation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 7, pp. 860–873, Jul. 2001.

[28] A. Khodakovsky, P. Schroder, and W. Sweldens, “Progressive geometry compression,” in *Proc. Comput. Graphics*, 2000, pp. 271–278.

[29] A. Lee, W. Sweldens, P. Schroder, L. Cowsar, and D. Dobkin, “MAPS: Multiresolution adaptive parameterization of surfaces,” in *Proc. SIGGRAPH’98*, 1998, pp. 95–104.

[30] Z. Yan, S. Kumar, J. Li, and C.-C. Kuo, “Robust encoding of 3D mesh using data partitioning,” in *Proc. 1999 IEEE Int. Symp. Circuits Systems*, vol. 4, 1999, pp. 495–498.

[31] H. Hoppe, “Progressive meshes,” in *Proc. ACM SIGGRAPH’96*, 1996, pp. 99–108.

[32] R. Ronford and J. Rossignac, “Full-range approximations of triangulated polyhedra,” in *Proc. Eurographics*, vol. 15, Aug. 1996, p. C–67.

[33] M. Garland and P. Heckbert, “Surface simplification using quadric error bounds,” in *Proc. ACM SIGGRAPH’97*, pp. 209–216.

[34] J. Rossignac, A. Safonova, and A. Szymczak, “3-D compression made simple: Edgebreaker on a corner-table,” in *Proc. Shape Modeling Int. Conf.*, Genoa, Italy, May 2001, pp. 278–283.

[35] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1983.

[36] B. Girod, K. Stuhlmuller, M. Link, and U. Horn, “Packet loss resilient internet video streaming,” in *Proc. SPIE Visual Commun. Image Process.*, San Jose, CA, Jan. 1999, pp. 833–844.

[37] U. Horn, K. Stuhlmuller, M. Link, and B. Girod, “Robust internet video transmission based on scalable coding and unequal error protection,” *Signal Process.: Image Commun.*, vol. 15, pp. 77–94, 1999.

[38] A. E. Mohr, E. A. Riskin, and R. E. Ladner, “Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 819–828, Jun. 2000.

[39] J. S. Choi, Y. H. Kim, H.-J. Lee, I.-S. Park, M. H. Lee, and C. Ahn, "Geometry compression of 3-D mesh models using predictive two-stage quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 312–322, Mar. 2000.

[40] W. Dong, J. Li, and C.-C. J. Kuo, "Progressive compression for triangular meshes with quality control," *Proc. SPIE*, vol. 4115, pp. 191–202, 2003.

Ghassan Al-Regib received the B.S. and M.S. degrees from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 1997 and 1998, respectively, both in electrical engineering, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 2003. His Ph.D. dissertation focused on developing error-resilient techniques to stream 3D graphics over lossy channels.

He is currently an Assistant Professor of electrical and computer engineering at Georgia Tech. While doing his doctoral studies, he worked for the Center for Signal and Image Processing (CSIP). He joined the faculty at Georgia Tech in August 2003, where he is currently working on projects related to multimedia communications, video and three-dimensional (3-D) graphics streaming, shared reality, multimodal processing of 3-D objects in shared spaces, and wireless sensor networks.

Dr. Al-Regib was the recipient of the Outstanding Graduate Teaching Award in 2000–2001 at the School of Electrical and Computer Engineering at Georgia Tech, the Center for Signal and Image Processing Research Award in 2003 at Georgia Tech, and the Center for Signal and Image Processing Service Award in 2003 at Georgia Tech

Yucel Altunbasak (S'94–M'97–SM'03) received the M.S. and Ph.D. degrees from the University of Rochester, Rochester, NY, in 1993 and 1996, respectively.

He is an Assistant Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta. He He joined Hewlett-Packard Research Laboratories in July 1996. Meanwhile, he taught at Stanford University, Stanford, CA, and San Jose State University, San Jose, CA, as a Consulting Assistant Professor. He joined the School of Electrical and Computer Engineering, Georgia Tech, in 1999. He is currently working on industrial- and government-sponsored projects related to multimedia networking, wireless video, video coding, genomics signal processing, and such inverse imaging problems as super-resolution and demosaicking. His research efforts to date resulted in over 100 peer-reviewed publications and 15 patents/patent applications. Some of his inventions have been licensed by the Office of Technology Licensing at Georgia Tech.

Dr. Altunbasak is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, *Signal Processing: Image Communications,* and the *Journal of Circuits, Systems and Signal Processing*. He served as the lead guest editor for the Special Issue on Wireless Video of *Signal Processing: Image Communications*. He has been elected to the IEEE Signal Processing Society's IMDSP Technical Committee. He has served as a co-chair for "Advanced Signal Processing for Communications" Symposia at ICC'03. He will serve as the technical program chair for ICIP-2006. He also served as a track chair at ICME'03, as a panel sessions chair at ITRE'03, as a session chair at seven international conferences, and as a panel reviewer for government funding agencies. He is a coauthor for a conference paper that received the Best Student Paper Award at ICIP'03. He received the National Science Foundation (NSF) CAREER Award in 2002, and he is a recipient of the 2003 Outstanding Junior Faculty Award at the School of Electrical and Computer Engineering, Georgia Tech.

Russell M. Mersereau (S'69–M'73–SM'78–F'83) received the S.B. and S.M. degrees and the Sc.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1969 and 1973, respectively.

He joined the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, in 1975. His current research interests are in the development of algorithms for the enhancement, modeling, and coding of computerized images, synthesis aperture radar, and computer vision. In the past, this research has been directed to problems of distorted signals from partial information of those signals, computer image processing and coding, the effect of image coders on human perception of images, and applications of digital signal processing methods in speech processing, digital communications, and pattern recognition. He is the coauthor of the text *Multidimensional Digital Signal Processing* (Englewood Cliffs, NJ: Prentice-Hall, 1984).

Dr. Mersereau has served on the Editorial Board of the PROCEEDINGS OF THE IEEE and as Associate Editor for Signal Processing of the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING and IEEE SIGNAL PROCESSING LETTERS. He was the Vice President for Awards and Membership of the IEEE Signal Processing Society and a member of its Executive Board from 1999 to 2001. He is the corecipient of the 1976 Bowder J. Thompson Memorial Prize of the IEEE for the best technical paper by an author under the age of 30, a recipient of the 1977 Research Unit Award of the Southeastern Section of the ASEE, and three teaching awards. He was awarded the 1990 Society Award of the Signal Processing Society and an IEEE Millenium Medal in 2000.