Turk J Elec Eng & Comp Sci, Vol.20, No.3, 2012, © TÜBİTAK
doi:10.3906/elk-1001-16

Joint source-channel coding for error resilient transmission of static 3D models

Mehmet Oğuz BİCİ[1,*], **Andrey NORKIN**[2], **Gözde AKAR**[1]
[1] *Middle East Technical University, Ankara-TURKEY*
e-mail: mobici@eee.metu.edu.tr
[2] *Tampere University of Technology, Tampere-FINLAND*

Received: 06.01.2010

Abstract

In this paper, performance analysis of joint source-channel coding techniques for error-resilient transmission of three dimensional (3D) models are presented. In particular, packet based transmission scenarios are analyzed. The packet loss resilient methods are classified into two groups according to progressive compression schemes employed: Compressed Progressive Meshes (CPM) based methods and wavelet based methods. In the first group, layers of CPM algorithm are protected unequally by Forward Error Correction (FEC) using Reed Solomon (RS) codes. In the second group, embedded bitstream obtained from wavelet based coding is protected unequally with FEC as well. Both groups of methods are scalable with respect to both channel bandwidth and packet loss rate, i.e. they try to optimize FEC assignments with respect to channel bandwidth and packet loss rates (PLR). In-depth analysis of these techniques are carried out in terms of complexity, robustness to losses and compression efficiency. Experimental results show that wavelet based methods achieve considerably better quality compared to CPM based methods.

Key Words: *Visual communications, error correction, computer vision, 3D models, wavelet transform, error resilience*

1. Introduction

With increasing demand for visualizing and simulating three dimensional (3D) objects in applications such as video gaming, engineering design, architectural walkthrough, virtual reality, e-commerce, scientific visualization and 3DTV, it is very important to represent 3D data efficiently [1] [2]. Among different representations, triangular 3D meshes are very effective and widely used. Typically 3D mesh data consists of geometry and connectivity data. While the geometry data specifies 3D coordinates of vertices, connectivity data describes the adjacency information between vertices. In this paper, we interchangeably use terms "3D model" and "3D mesh."

To maintain a convincing level of realism, many applications require highly detailed complex models represented by 3D meshes consisting of huge number of triangles. Due to storage space and transmission bandwidth limitations, there has been a great deal of research effort into the efficient compression of 3D meshes

[*] Corresponding author: Middle East Technical University, Ankara-TURKEY

[1] [2]. On the other hand, transmission of 3D meshes over error-prone channels where packets may be lost or delayed because of congestion, buffer overflow, uncorrectable bit errors or misrouting is not tackled at the same level.

The pioneering work in error resilient 3D model streaming is that of Bajaj et al. [3] where compressed VRML streaming problem is addressed. In this method, the encoded bitstream is classified into independent layers according to the depth-first order of the vertices. In this way, a layer can be decoded regardless of whether other layers are received or not. In [4], error resilience is achieved by segmenting the mesh and transmitting each segment independently. At the decoder, these segments are stitched using the joint-boundary information which is considered the most important. Drawbacks of these algorithms are that they are not scalable with respect to the channel packet loss rate, P_{LR} and they do not provide a coarse-to-fine representation of the model.

In [5], [6], transmission of 3D objects represented by texture and mesh over unreliable networks is described. For arbitrary meshes, stripification of the mesh and distributing nearby vertices into different packets are considered, combined with a strategy that does not need texture or mesh packets to be retransmitted, with the exception that only the valence (connectivity) packets need to be retransmitted. Lost geometry is interpolated and different interpolation strategies are evaluated for this algorithm in case of losses. Each packet is compressed independently to support loss resilience at the expense of decreasing compression efficiency with increasing number of packets. However, no optimization is done with respect to channel characteristics.

In [7], an error resilient packetization scheme is proposed for 3D models with the motive of decreasing the dependencies among packets. To model the packetization, first a Non-Redundant Directed Acyclic Graph (DAG) is constructed to encode the dependencies among the vertex splits of a progressive mesh. A special Global Graph Equipartition Packing Algorithm is then applied to partitioning this graph into several equal size sub-graphs, which is packed as packets. However, there is again no optimization with respect to channel characteristics and the method requires retransmission.

Multiple Description Coding (MDC) is also used to achieve error resiliency in [8], [9], [10]. In [8], multiple descriptions are generated by splitting the mesh geometry into submeshes and including the whole connectivity information in each description. In [9], multiple description scalar quantization (MDSQ) is applied to wavelet coefficients of a multiresolution compression scheme. The obtained two sets of coefficients are then independently compressed by the SPIHT (*Set Partitioning in Hierarchical Trees*) coder [11]. In [8] and [9], descriptions are created with heuristic methods and no optimum solutions are proposed for varying network conditions. In [10], wavelet coefficient trees obtained by Progressive Geometry Compression (PGC) [12] algorithm are partitioned into multiple descriptions. Each set of trees is independently coded with SPIHT. In this scheme, bit-rate for each set is optimized for a given description loss rate. The MDC schemes mentioned above are resilient to description losses. This is useful in scenarios such as multipath transmission or multiple storage. However, the MDC schemes are not directly applicable to packetized streaming/transmission scenarios where the packet sizes and description sizes differ considerably.

In [13], [14], [15], [16], [17] error resilient techniques that are scalable with respect to both channel bandwidth and packet loss rate (P_{LR}) are proposed. The methods in [13], [14], [15], [16] try to achieve error resilience by assigning optimal *Forward Error Correction* (FEC) codes to layers of a progressively coded 3D mesh. The progressive scheme employed in these works is *Compressed Progressive Meshes* (CPM) [18]. While the ideas are similar in these works, a more general optimization problem is tackled in [14] which maximizes expected decoded model quality for a given model, total bit budget and P_{LR}. Another important property of these methods is that the 3D model can be reconstructed at a resolution between coarse and fine representation with respect to varying packet loss rates.

On the other hand, in [17], a wavelet based method for robust transmission of 3D models in a packet loss network is proposed. The proposed algorithm depends heavily on FEC based packet lost resilient image transmission schemes [19]. The method is based on obtaining an embedded bitstream from wavelet based *Progressive Geometry Compression* (PGC) [12] and protecting the embedded bitstream with optimal FEC with respect to channel bandwidth and P_{LR}.

In this work, we address error resilient techniques that are scalable with respect to both channel bandwidth and packet loss rate, i.e. CPM based and wavelet based methods. Our purpose is to give in depth analysis of these techniques in terms of complexity, robustness to losses, compression efficiency and suitability for real time transmission using a common measure and also propose improvements. Even though each of these methods are addressed in the literature, to our knowledge, there is no study that performs such an analysis. In the first group, we compare CPM based methods and propose modifications to decrease the complexity, using both original L^2 error and quadric error metric. In this way we manage to do performance evaluation for 3D models with high number of triangles. Then, we compare CPM based methods with the wavelet based methods in terms of average distortion of received model, complexity of optimization. The experimental results show that, higher quality with more flexible packetization can be achieved by the proposed algorithm.

The remainder of this paper is organized as follows. Section 2 describes the CPM and wavelet based loss resilient 3D mesh coding techniques and proposes improvements to CPM based methods. Section 3 provides overview of the distortion metric used for 3D meshes and ways of reducing computation times for optimization. Section 4 describes the channel model employed in experiments. Experimental results are presented in Section 5 and, finally, we conclude in Section 6.

2. Loss resilient 3D mesh coding

Performance of loss resilient coding techniques is highly correlated with the compression techniques on which they are based. 3D mesh compression techniques can be classified into two categories: Single-rate compression and Progressive compression. In single-rate compression, the aim is to compress the mesh as much as possible. The single-rate compressed mesh can only be decompressed if the whole compressed bitstream is available, i.e. no intermediate reconstruction is possible with fewer bits. Progressive compression is more suited for transmission purposes in which some parts of the compressed bitstream can be missing or erroneous. By progressive compression, the mesh is represented by different levels of detail (LODs) having different sizes. Progressive compression techniques can further be classified into two categories: connectivity driven and geometry driven compression. In connectivity driven progressive mesh compression schemes, the compact representation of connectivity data is given a priority and geometry coding is driven by connectivity coding. On the other hand, in geometry driven compression data is compressed with little reference to connectivity data, i.e. even the mesh connectivity can be changed in favor of a better compression of geometry data. It is shown in [2] that better compression ratios can be obtained by geometry driven progressive compression methods.

In this section, we analyze two loss resilient 3D mesh coding approaches that are classified according to the progressive mesh compression schemes employed, namely *Compressed Progressive Meshes* (CPM) [18] which belong to the connectivity driven schemes and wavelet based *Progressive Geometry Compression* (PGC) [12] which belong to the geometry driven schemes. Since these works are the best representatives of each category, we carry out the analysis based on them.

2.1. Compressed progressive meshes-based loss resilient coding

2.1.1. Compressed progressive meshes

As noted earlier, the CPM method is a connectivity driven progressive mesh compression scheme. Therefore if the whole bitstream is received successfully, then the original connectivity of the model can be reconstructed. The encoder starts with the original mesh and generates meshes at different LODs iteratively. During each iteration a simplified and coarser model LOD is generated from the present LOD of the model.

The basic operation for coarsening the present LOD is the edge collapse operation. This operation combines two vertices of an edge into one vertex by collapsing the edge. This results in a decrease in the number of triangles by two. The destroyed triangles form the cut-edges that are incident on the newly generated vertex. In each iteration, a certain subset of edges are chosen to be collapsed. The encoder decides to stop generating coarser LODs at a point and ends up with the simplest base mesh and M LODs.

The decoder performs in the reverse direction of encoder. It starts with the base mesh and constructs finer LODs in each iteration. The basic operation for this construction is the vertex split operation. This operation produces two new vertices from the vertex that was generated by collapsing an edge in the encoder. The locations of new vertices are predicted and displacement errors are corrected. The details of the levels increase in each iteration as new triangles are generated from the cut-edges.

All the operations needed for the decoder to decode a finer level from the present level in an iteration is coded as a batch in the encoder. The encoded batch bitstream is composed of (a) a *Collapse Status*, one bit to specify whether a vertex is to be split or not; (b) *Cut Edges*, the indices of cut-edges for the vertices to be split; and (c) *Position Error*, quantized and entropy coded difference in geometric coordinates between the collapsed vertex and the predicted vertex locations. Compressing the base mesh with a single-rate coder, the final bitstream of the CPM algorithm is generated by the concatenation of compressed bitstream of base mesh (base layer) of size $R^{(0)}$ and the M batches (M enhancement layers) of size $R^{(i)}$, $i = 1, \ldots, M$.

2.1.2. Loss resilient coding

For error resilient transmission of the data generated by the CPM algorithm, optimal error correcting codes (in particular, Reed-Solomon (RS) codes) can be assigned to the layers of the progressively coded mesh. Let $RS(N, k_0)$ denote the RS code applied to base layer (level-0) and $RS(N, k_j)$ denote the RS code applied to j-th enhancement layer (level-j), where $j = 1, \ldots, L_M$. Here, L_M denotes the number of enhancement layers transmitted out of M enhancement layers according to the bitrate of the channel. RS codes are applied vertically and packetization is performed horizontally. Therefore, receiving any k_j out of N packets allows successful decoding of level-j. A simple protected CPM output bitstream is illustrated in Table 1.

The problem definition can be formulated as follows. Given a 3D model and a total bit budget B, the aim is to determine an optimal combination of the following parameters to minimize the expected decoded model distortion ($E_{\mathrm{D}}(L_M)$): (a) a value l, for the number of bits used in quantizing the position error; (b) L_M, the number of transmitted batches; (c) C, the total number of channel coding bits; and (d) $C_L = [C^{(0)}, C^{(1)}, \ldots, C^{(L_M)}]$, in which $C^{(i)}$ denotes the number of channel coding bits applied to level i (or $[k_0, k_1, \ldots, k_{L_M}]$, since k_i is a function of $C^{(i)}$, $R^{(i)}$ and N).

To quantify the expected distortion $E_{\mathrm{D}}(L_M)$, first let P_j denote the probability of terminating the

Table 1. An example CPM output with $L_M + 1 = 3$ layers. Pi denotes Packet i generated horizontally, while FEC is applied vertically. In this simple example, $N = 6$, $k_0 = 2$, $k_1 = 3$ and $k_2 = 4$.

Base Mesh			Batch 1			Batch 2			
P1	1	2	P7	5	6	P13	11	12	13
P2	3	4	P8	7	8	P14	14	15	16
P3	FEC	FEC	P9	9	10	P15	17	18	19
P4	FEC	FEC	P10	FEC	FEC	P16	20	21	22
P5	FEC	FEC	P11	FEC	FEC	P17	FEC	FEC	FEC
P6	FEC	FEC	P12	FEC	FEC	P18	FEC	FEC	FEC

decoding operation at level-j and it can be calculated as

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

where $p(m, N)$ is the probability of losing m packets within a block of N packets. Then the expected distortion $E_{\mathrm{D}}(L_M)$ can be calculated as

$$E_{\mathrm{D}}(L_M) = P_0 D_{NR} + \sum_{j=1}^{L_M} P_j D_{j-1} \prod_{i=0}^{j-1}(1 - P_i) + D_{L_M} \prod_{j=0}^{L_M}(1 - P_j), \tag{2}$$

where D_j is the distortion of level-j, D_{NR} is the distortion when no reconstruction of the model is possible (i.e., if the base mesh is lost), D_{L_M} is the distortion if all the levels are successfully received.

2.2. Proposed modifications for CPM based loss resilient coding

In [14], Al-Regib et al. have proposed an algorithm to find the optimum solution for this problem. The algorithm is based on selecting the best $(l,\ L_M)$ pair for each value of C. Since it is not feasible to change C bit by bit, the unknown variable C is quantized by a step size Q. In each step the best $(l,\ L_M)$ pair is selected using rate-distortion (RD) curve. For each selected l and L_M in that step, best expected distortion and C_L are found using a local search algorithm. Final output of the algorithm is the C_L corresponding to best distortion among all steps.

A drawback of this algorithm is that it contains many repeated operations since the results are iterated by varying C using the step size Q. As there are finite choices of l, L_M and k_j values for a C, it is very likely to encounter same l, L_M and k_j values for several C values during the local search. In [13], this computational redundancy is removed by iterating only the finite k_j values and putting the constraint that $k_0 \leq k_1 \leq \cdots \leq k_L$. Although in [13] the problem definition also states that C is given and L_M is fixed, we can generalize the algorithm by removing the assumptions that C and L_M are fixed. In our experimental results, we propose to generalize the problem by combining the methods of [14] and [13] such that, for given possible finite sets of l and L_M,

- The combined algorithm computes expected distortion for every k_j values satisfying the bit budget requirement and the condition $k_0 \leq k_1 \leq \cdots \leq k_L$.

- k_j values corresponding to the least expected distortion is chosen as the optimum FEC assignment.

In [15], Al-Regib et al. tackle with a similar problem in which l is also assumed to be given in addition to our general problem definition. An exhaustive search of $[C^{(0)}, C^{(1)}, \ldots, C^{(L)}]$ is proposed to find the optimal solution. In [16], Ahmad et al. propose improvements on [15] in terms of complexity and packetization flexibility inspired from the work in [20]. As this algorithm assumes that l is given, we propose a minor modification to handle the general problem definition. The modified algorithm is such that, first the local optimum parameters are found for a set of l values and the parameters corresponding to the best expected distortion is chosen as the global optimum parameters. The experimental results are obtained using the modified algorithm.

2.3. Wavelet based loss resilient coding

2.3.1. Wavelet based scalable mesh coding

As noted earlier, wavelet based mesh coding techniques belong to the geometry driven progressive mesh coding category. Very efficient wavelet based compression schemes have recently been reported in literature [12], [21], [22], [23], [24] and have inspired the wavelet subdivision surfaces tool of MPEG-4's Animation Framework eXtension (AFX) [25],[26].

In this paper, we use PGC scheme [12] to produce scalable bitstream since this scheme represents the basic idea behind all wavelet based codecs. The other wavelet based compression schemes [21],[22] can also be used with minor modifications. PGC is a progressive compression scheme for arbitrary topology, highly detailed and densely sampled meshes arising from geometry scanning. The method is based on smooth semi-regular meshes, i.e., meshes built by successive triangle quadrisection starting from a coarse irregular mesh. Therefore the original model in PGC should be remeshed [27] to have a semi-regular structure which allows subdivision based wavelet transform. Resulting semi-regular mesh undergoes a loop-based [28] or butterfly-based [29] wavelet decomposition to produce a coarsest level mesh and wavelet coefficients [12]. Since coarsest level connectivity is irregular, it is coded by Touma and Gotsman's (TG)[30] single-rate coder. Wavelet coefficients are coded with *Set Partitioning in Hierarchical Trees* (SPIHT) algorithm [11]. For improved progressivity, a predetermined number of bit-planes of the coarsest level geometry can be transmitted initially with the coarsest level connectivity. The remaining refinement bit-planes can be transmitted as the SPIHT coder descends a given bit-plane of wavelet coefficients [12]. As a result, an embedded bitstream is generated as illustrated in Figure 1.

2.3.2. Loss resilient coding

After the embedded bitstream is defined, the problem of optimum loss protection is stated as follows [17]: The embedded bitstream is to be protected with RS codes and transmitted over an erasure channel as N packets, each of which contains L symbols (bytes, in this paper). The protection system builds L source segments S_i of m_i symbols each where $i = 1, \ldots, L$ and $m_i \in \{1, \ldots, N\}$ and protects each segment with an $RS(N, m_i)$ code. For each $i = 1, \ldots, L$, let $f_i = N - m_i$ denote the number of RS redundancy symbols that protect segment S_i. An example of the above FEC assignment is illustrated in Table 2. If n out of N packets are lost, then the RS codes ensure that all segments that contain at most $N - n$ source symbols can be recovered. Thus, by adding the constraint that $f_1 \geq f_2 \geq \cdots \geq f_L$, if at most f_i packets are lost, then the receiver can decode at least the

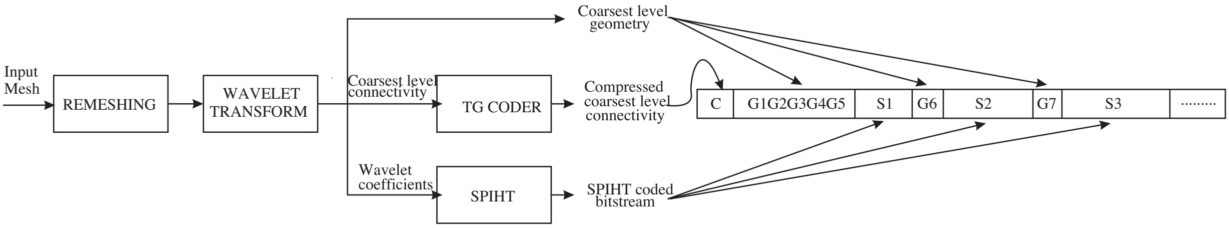


Figure 1. Diagram showing generation of the embedded bitstream from a PGC coder. The bitstream starts with compressed coarsest level connectivity C, as it is the most important part on which the whole mesh connectivity depends. The next part of the bitstream is a predetermined number of bit-planes (5 shown in the figure) of the coarsest level geometry (*G1 G2 G3 G4 G5*) since wavelet coefficients would have no use without the coarsest level geometry. The remaining part of the bitstream consists of the output bitstream of SPIHT for different quantization levels (*S1 S2 S3...*) and after each quantization level, refinement bit-planes of coarsest level geometry (*G6 G7...*) are inserted for improved progressivity.

first i segments. Let $\mathcal{F}$ denote the set of L-tuples $(f_1, \ldots, f_L)$ such that $f_i \in \{0, \ldots, N-1\}$ for $i = 1, \ldots, L$ and $f_1 \geq f_2 \geq \cdots \geq f_L$. Let $p(m, N)$ denote the probability of losing exactly m packets of N and let

$$c_N(k) = \sum_{m=0}^{k} p(m, N) \text{ for } k = 0, \ldots, N. \tag{3}$$

Then $c_N(f_i)$ is the probability that the segment S_i can be decoded successfully.

Let $D(R)$ denote the distortion-rate (D-R) function of the source coder. Then in order to achieve an optimum the packet loss protection, it is needed to find $F = (f_1, \ldots, f_L) \in \mathcal{F}$ such that the expected distortion

$$E_{\mathrm{D}} = c_N(N)D(r_0) + \sum_{i=1}^{L} c_N(f_i)(D(r_i) - D(r_{i-1})) \tag{4}$$

is minimized, where

$$r_i = \begin{cases} 0, & \text{for } i = 0 \\ \sum_{k=1}^{i} m_k = iN - \sum_{k=1}^{i} f_k, & \text{for } i = 1, \ldots, L. \end{cases} \tag{5}$$

The next step is to determine the optimum FEC assignments by minimizing E_{D} in equation 4. In the literature, there are several efficient methods for similar optimization problems used for scalable image coders [19], [31], [32], [33], [34], [20]. The CPM based methods can also be used for FEC assignment assuming that partitions of an embedded bitstream are batches of a progressive coder. However, since embedded coders are highly progressive, efficient optimization methods are needed other than limited iteration steps used in CPM.

In [34], it is shown that the method in [31] performs very well in terms of expected distortion and the method in [34] has the lowest computational complexity with slightly worse expected distortion performance.

In [31], given $p = LN$ points on the operational D-R curve of the source coder, the algorithm first computes the h vertices of their convex hull. Then a solution is found in $O(hN \log N)$ time. This solution is optimal under the assumption of the convexity of the D-R function and of fractional bit allocation assignment. In [34], a local search algorithm with $O(NL)$ complexity is presented that starts from a solution that maximizes the expected number of received source bits and iteratively improves this solution. The reader is referred to

[31], [34] for the details of the algorithms. Since we also use a scalable bitstream produced by PGC coder, we employ the optimization methods in [31] and [34] in our experiments.

Table 2. A table showing an example of FEC assignment. There are $N = 5$ packets, each composed of $L = 4$ symbols. as such, there are 4 source segments, S_i, $i = 1, 2, 3, 4$, each of which contains m_i data symbols and f_i FEC symbols, where $m_i + f_i = N$. In this example, $m_1 = 2$, $f_1 = 3$, $m_2 = 3$, $f_2 = 2$, $m_3 = 3$, $f_3 = 2$, $m_4 = 4$ and $f_4 = 1$. Earlier parts of the bitstream are assigned more FEC symbols since they contribute more to overall quality.

	P1	P2	P3	P4	P5
Segment 1	1	2	FEC	FEC	FEC
Segment 2	3	4	5	FEC	FEC
Segment 3	6	7	8	FEC	FEC
Segment 4	9	10	11	12	FEC

3. Distortion metric and simplifications in calculations

In order to use equations 2 and 4, we need to quantify quality of the model, i.e. distortions D_j and $D(R)$ in the equations. To be able to describe the quality of a processed/reconstructed 3D model, either objective or subjective quality measures need to be defined. There is no immediate objective distortion metric in 3D meshes like mean-square error in images. One distortion metric used in the literature is the Hausdorff distance $d_{\mathrm{H}}(X, Y)$ between two surfaces X and Y, defined by

$$d_{\mathrm{H}}(X,Y) = \max\{\max_{x\in X} d(x,Y),\, \max_{y\in Y} d(y,X)\}, \tag{6}$$

where $d(x, Y)$ is the Euclidean distance from a point x on X to the closest point on Y. Another distortion metric is the L^2 distance, $d_L(X, Y)$, between two surfaces X and Y and is defined by

$$d_L(X,Y) = \max\{ d(X,Y),\, d(Y,X) \} \tag{7}$$

where

$$d(X,Y) = \left(\frac{1}{\mathrm{area}(X)} \int_{x\in X} d(x,Y)^2 dx \right)^{1/2}. \tag{8}$$

Hausdorff distance takes the maximum of Euclidean distances, whereas L^2 distance (Euclidean length) involves the root mean square of the distances. Therefore, L^2 distance reflects the average distortion of a 3D model while the Hausdorff distance reflects the maximum error. For this reason we use L^2 distance to denote the objective distortion metric in the experiments. We use Metro tool [35] to compute this distance.

As the L^2 distance has an expensive computation cost, considerable offline computations can be required during optimization. To reduce this complexity, some simplifications need to be employed. For CPM based methods, quadric error metric can be used and, for wavelet based methods, distortion rate curve modeling can be employed, as explained in following two sections.

3.1. Simplification by quadric error metric

In both Al-Regib and Ahmad's works, complete D-R curves are required for each quantization level during optimization. But since the calculation of L^2 distance is not an inexpensive operation, obtaining D-R curves

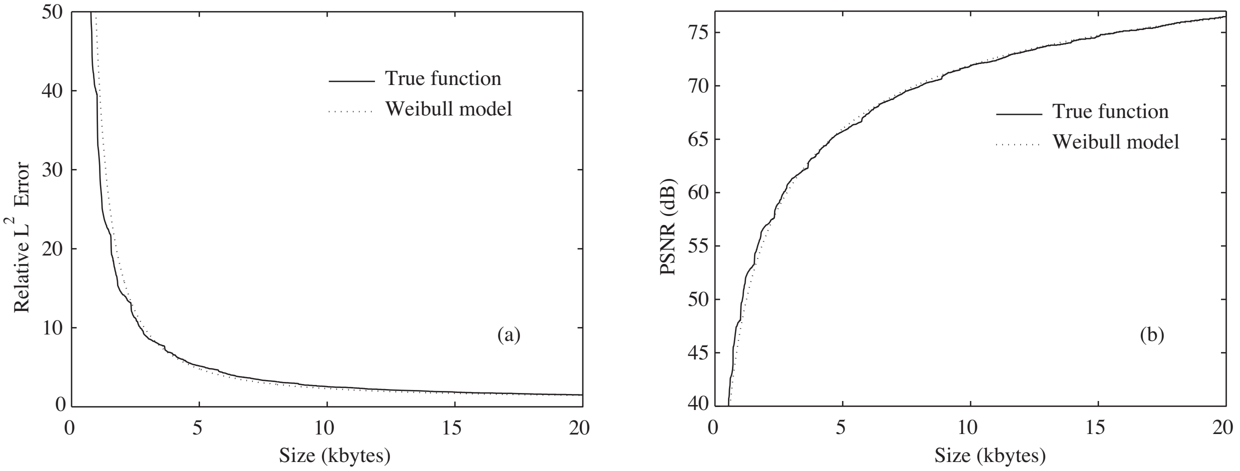


Figure 2. Comparison between the Weibull model (10 points) and operational RD curve (L^2) for *Bunny* model. (a) Relative L^2 error; (b) PSNR.

can be a very long process. In order to account for this problem, quadric error metric can be used during optimization as proposed in [13]. In this way, the distortion is estimated by using a conservative upper bound on the maximum distortion that results from edge collapses in a batch, which might be obtained using the sum of the squares 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 [18]. Since CPM coder inherently uses this metric during compression, it does not have an extra time cost at all.

3.2. Simplification by distortion rate curve modeling

In wavelet based loss resilient mesh coding, there are many rate values where the mesh can be reconstructed due to embedded bitstream. Calculation of the whole distortion rate (D-R) curve requires considerable time. Therefore we employ the Weibull modeling of D-R curve presented in [36] that is used for coding of images. It is found in [10] that output of PGC coder can also be approximated with this model. The model is described by

$$D(R) = a - be^{-cR^d}, \tag{9}$$

where real numbers a, b, c, and d are the parameters which depend on the D-R characteristics of the source and the bitstream. As there are four parameters in this model, $D(R)$ curve can be found by using at minimum four points. This model can approximate both L^2 and PSNR curves where $PSNR = 20\log_{10} 1/L^2 error$. To fit this model to RD samples, we use nonlinear least-squares regression.

Figure 2 shows a comparison of true operational D-R curves of *Bunny* model and their Weibull models. The Weibull models are $D(R) = 1283.24 - 1283.24\,e^{-0.158R^{-1.550}}$ and $D(R) = 82.78 - 142.05\,e^{-0.216R^{0.269}}$ for L^2-distance and PSNR, respectively.

One can see that the model closely approximates the real data. Moreover, the model has a nice feature of convexity, which is desirable for bit allocation algorithm.

4. Channel model

In order to consider packet loss behavior to optimize expected decoded model quality, the channel is needed to be modeled with an appropriate model. In this paper we use the two state Markov model [37] for packet losses. This model is investigated and shown to be very effective to model packet losses in [38] and [39].

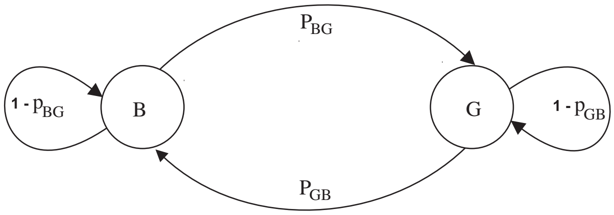


Figure 3. Two state Markov channel model [37].

In this section, we briefly review the model and the reader is referred to [38] and [39] for more details. The Markov model described in [38] and [39] is a renewal model, i.e., the event of a loss resets the memory of the loss process. Such models are determined by the distribution of error free intervals (gaps). Let a gap of length v be the event that after a lost packet $v-1$ packets are received and then again a packet is lost. The gap density function $g(v)$ gives the probability of a gap length v, i.e., $g(v) = \Pr(0^{v-1}1|1)$, where '1' denotes a lost packet and '0^{v-1}' denotes $v-1$ consecutively received packets. The gap distribution function $G(v)$ gives the probability of a gap length greater than $v-1$, i.e., $G(v) = \Pr(0^{v-1}|1)$. In our model, in state B all packets are lost ('1'), while in state G all packets are received ('0'), yielding

$$g(v) = \begin{cases} 1 - p_{\text{BG}}, & \text{if } v = 1, \\ p_{\text{BG}}(1 - p_{\text{GB}})^{v-2} p_{\text{GB}}, & \text{if } v > 1, \end{cases} \tag{10}$$

$$G(v) = \begin{cases} 1, & \text{if } v = 1, \\ p_{\text{BG}}(1 - p_{\text{GB}})^{v-2}, & \text{if } v > 1. \end{cases} \tag{11}$$

Let $R(m, N)$ be the probability of $m-1$ packet losses within the next $N-1$ packets following a lost packet. It can be calculated using the recurrence

$$R(m, N) = \begin{cases} G(N), & m = 1, \\ \sum_{v=1}^{N-m+1} g(v) R(m-1, N-v), & 2 \leq m \leq N. \end{cases} \tag{12}$$

Then the probability of m lost packets within a block of N packets given in equations 3 and 1 is

$$p(m, N) = \sum_{v=1}^{N-m+1} P_B G(v) R(m, N-v+1), \text{if } 1 \leq m \leq N, \tag{13}$$

where P_B is the average loss probability.

5. Experimental results

We have performed the experiments with *Bunny* and *Venus head* models. *Bunny* model is composed of 34835 vertices and 69472 triangles and *Venus head* model is composed of 50002 vertices and 100000 triangles. The original uncompressed models are shown in Figure 4. The simulated distortions given in the results are calculated by averaging 100 experiments of channel simulation. The packets are sent over a packet erasure channel which is modeled as two-state Markov process with average burst length of 5. The reconstruction distortion is relative L^2 error, which is calculated by Metro tool[35]. Relative error is calculated by dividing L^2 distance to the original mesh bounding box diagonal. All the relative L^2 errors in this paper are in units of 10^{-4}. We also provide the same numbers in PSNR scale where $PSNR = 20\log_{10} \text{peak}/d$, where *peak* is the bounding box diagonal, and d is the L^2 error. The bitrate values are presented in terms of bits per vertex (bpv).

(a)

(b)

Figure 4. Images of the test data used in this work: (a) *Bunny* model; (b) *Venus head* model.

For computation complexity experiments, we measure the time taken for an algorithm to run with an Intel Pentium 4, 2.2 GHz, 1 Gb RAM, Windows XP installed computer. Although it is not possible to achieve exact complexity results with these settings (e.g. due to possible inefficient implemented parts of algorithms or multitasking of OS), we provide the results to mention the order of complexity of the algorithms compared to each other.

In the following experimental results, if the 3D model name and the coding bitrate are not explicitly specified, then these results correspond to the *Bunny* model coded at 3.5 bpv and packetized with $N = 100$.

In the rest of this section, we categorize the experiments and present in different subsections. We start with the results of CPM based methods which include the proposed *kStep* parameter and comparison of CPM based method simulation distortions. Then we provide the results for D-R curve modeling for PGC based methods followed by the comparison of CPM and PGC based methods in terms of simulation distortions. Then we examine the mismatch scenario which occurs when the real loss rate and the one used in optimization differ. We continue the results with complexity comparisons for the optimization methods and finally we provide visual results for subjective evaluation.

5.1. Proposed kStep for CPM based methods

For Al-Regib's CPM based methods, since iterating all possible $RS(N, k_j)$ pairs for each layer is not feasible due to significant complexity requirements, we propose a new parameter, $kStep$. With parameter $kStep$, instead of iterating k_j's in $RS(N, k_j)$ pairs one by one, we increment and decrement k_j values by an amount of $kStep$ in the iterations. Figure 5 shows the simulated PSNR values and Figure 6 shows the optimization times for different $kStep$ values.

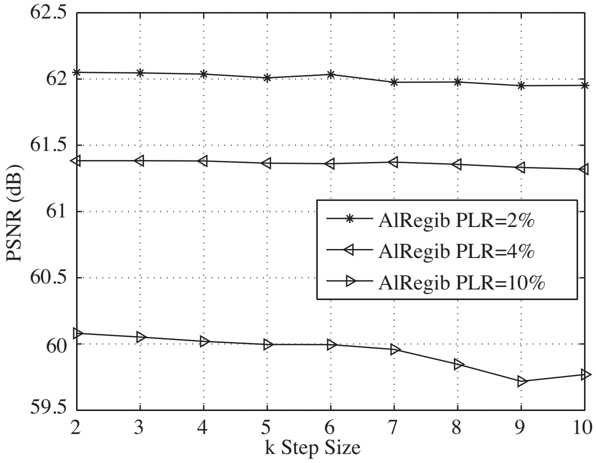


Figure 5. Graph showing effect of step size k ($kStep$) on quality: Simulated PSNR vs. step size k for *Bunny* model, optimized for $P_{\mathrm{LR}} = 2\%$, 4% and 10%, coded at 3.5 bpv.

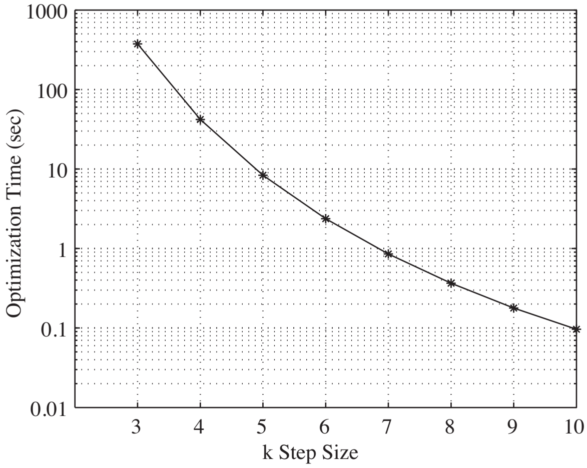


Figure 6. Graph showing effect of step size k ($kStep$) on complexity: Optimization time vs. step size k for *Bunny* model optimized for $P_{\mathrm{LR}} = 4\%$ and coded at 3.5 bpv.

From the figures, it can observed that it is possible to save great amount of time during optimization by increasing the $kStep$ value. In addition, while increasing the $kStep$ value, the decrease in the simulated PSNR value is not significant for different packet loss rates.

5.2. Comparison of CPM based methods

After defining the *kStep* parameter, we proceed to comparison of all aforementioned CPM based methods. Figure 7 summarizes all of the mentioned CPM based error resilient methods. In the figure, simulated PSNR vs P_{LR} values are presented for several configurations. The curves in the figure are Al-Regib's method with a fine *kStep* value of 2 (*Al-Regib kStep = 2*), Al-Regib's method with a coarser *kStep* value of 5 (*Al-Regib kStep = 5*), Al-Regib's method with *kStep* = 5 and using quadric error metric during optimization (*Al-Regib kStep = 5 Quad*), generalized Ahmad's method (*Ahmad*) and generalized Ahmad's method with using quadric error metric during optimization (*Ahmad Quad*).

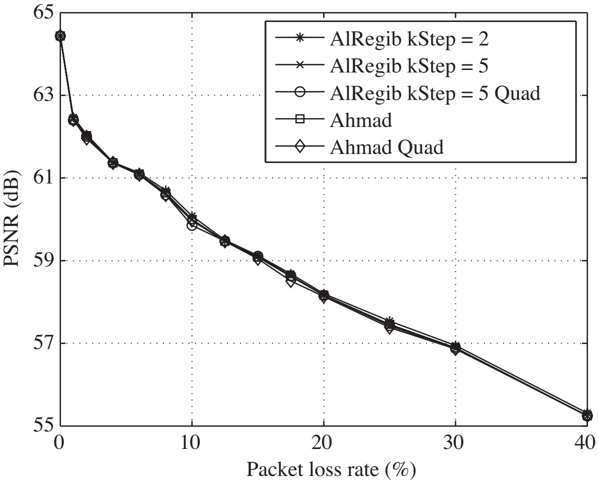


Figure 7. Comparison of CPM based methods for Bunny model in terms of simulated distortions for various P_{LR}'s.

From the figure, it can be deduced that all the CPM based methods similar performance, where none of the methods achieve significantly higher simulated PSNR. Nonetheless, best PSNR is achieved by *Al-Regib kStep = 2* expectedly.

5.3. Performance of D-R Curve modeling for PGC based methods

After examining the CPM based methods, we start the analysis of PGC based methods. For the PGC based methods, FEC assignments are optimized with the algorithms of Mohr et al. and Stankovic et al. [31], [34] and labeled as *PGCMohr* and *PGCStankovic* in the rest of the paper.

We initially examine the performance of the proposed D-R curve modeling described in Section 3.2. Figure 8 shows simulated distortions corresponding to various P_{LR}'s for *PGCMohr* employing the original D-R curve and modeled D-R curve during optimization. It is observed that quite acceptable results can be achieved by D-R curve modeling. Therefore we present the remaining PGC based results with modeled D-R curves which significantly reduce optimization times.

5.4. Comparison of CPM and PGC based methods

In this part, we present comparison of the CPM and PGC based methods in terms of simulation distortions for three cases: 1) Bunny Model Coded at 3.5 bpv, 2) Bunny Model Coded at 1.2 bpv and 3) Venus Model Coded at 4 bpv.

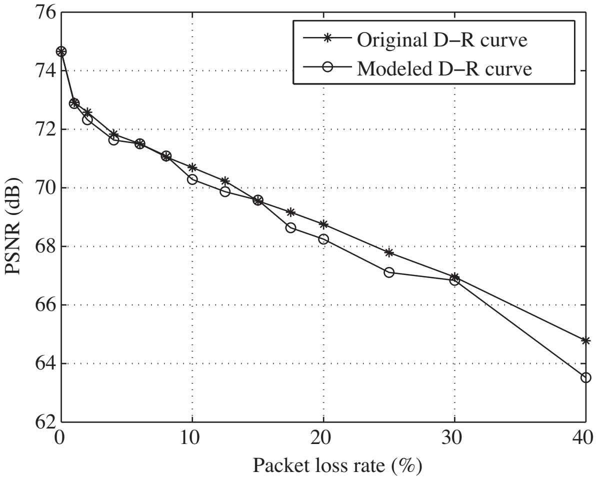


Figure 8. Comparison of using original D-R curve and using modeled D-R curve during optimization for *Bunny* model in terms of simulated distortions for various P_{LR}'s.

5.4.1. Bunny model coded at 3.5 bpv

For this case, a comprehensive summary of the results for all methods is presented in Table 3. Also, comparison of CPM based and PGC based methods in terms of simulated PSNR can be seen in Figure 9. From the results, one can notice that PGC based method significantly outperform the CPM based methods.

Table 3. Simulated distortion results of the first scenario for various P_{LR} values. The distortion metric is relative L^2 error in units of 10^{-4}.

	Simulated distortion for different P_{LR}								
	PLR = 0%	PLR = 2%	PLR = 4%	PLR = 6%	PLR = 10%	PLR = 15%	PLR = 20%	PLR = 30%	PLR = 40%
PGCStankovic	1.89	2.52	2.61	2.78	3.09	3.48	3.78	4.7	5.89
PGCmohr	1.85	2.42	2.62	2.66	3.06	3.32	3.87	4.55	6.67
Al-Regib kStep = 2	6	7.8979	8.5275	8.7846	9.9071	11.0633	12.3032	14.2119	17.1574
Al-Regib kStep = 5	6	7.9356	8.5464	8.8324	10.0043	11.0837	12.355	14.32	17.2934
Al-Regib kStep = 5 Quad	6	7.9362	8.5464	8.8324	10.1721	11.0837	12.3602	14.32	17.2934
Ahmad	6	7.92	8.5411	8.8165	10.0053	11.113	12.3531	14.3025	17.2771
Ahmad Quad	6	7.9823	8.5455	8.8319	10.0355	11.1606	12.3983	14.3486	17.2771

5.4.2. Bunny model coded at 1.2 bpv

In this case, we decrease the bitrate and code Bunny model at 1.2 bpv to observe low bitrate error resilient characteristics. Comparison of PGC and CPM based methods is provided in Figure 10. We observe that the results do not change by decreasing the coding bitrate.

5.4.3. Venus model coded at 4 bpv

Finally in this case, we repeat the experiments performed on *Bunny* model with *Venus head* model which is coded at 4 bpv and packetized with $N = 100$. Simulated PSNR comparison of PGC based and CPM based methods can be seen in Figure 11. Similar to the results with the *Bunny* model, we observe that PGC based methods, again, significantly outperform the CPM based methods.

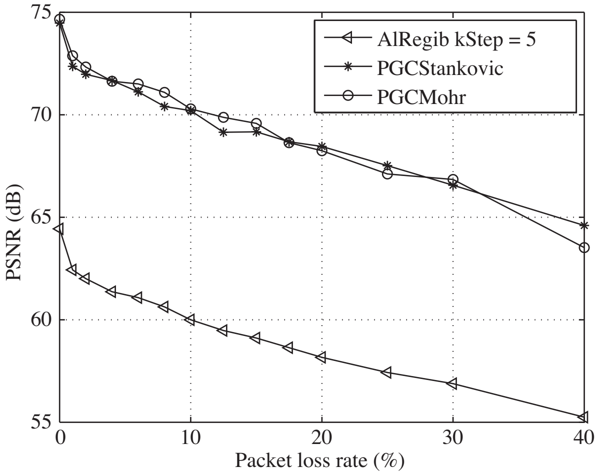


Figure 9. P_{LR} vs. Simulated distortion in PSNR scale for the *Bunny* model coded at 3.5 bpv.

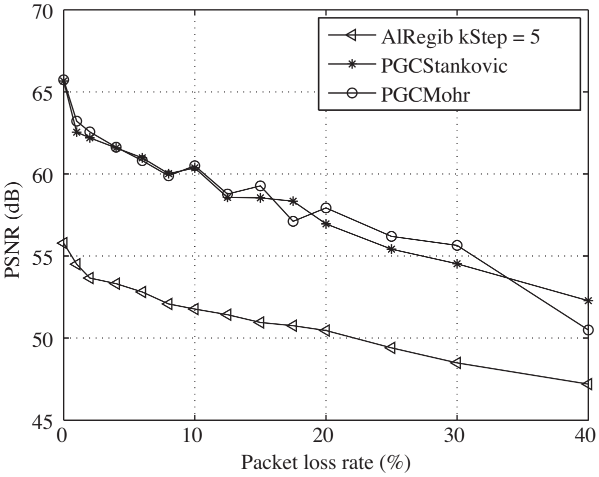


Figure 10. P_{LR} vs. Simulated distortion in PSNR scale for *Bunny* model coded at 1.2 bpv.

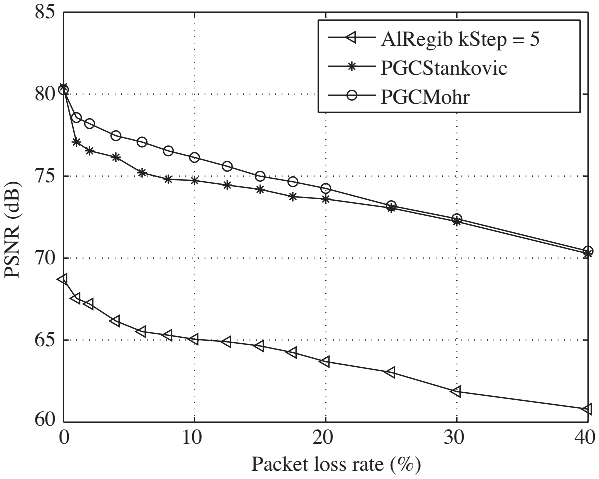


Figure 11. P_{LR} vs Simulated distortion in PSNR scale for *Venus head* model coded at 4 bpv.

5.5. Mismatch cenario

In the experiments presented so far, the assumption is that, during the optimization, we know the channel loss rate and optimize the protection parameters accordingly. However, in real cases, the channel packet loss rate used during the optimization and the actual loss rate encountered may differ. Therefore in this part, we investigate what happens when a model optimized for a loss rate is transmitted over a channel with a different lost rate. The results of this experiment are shown in Figure 12.

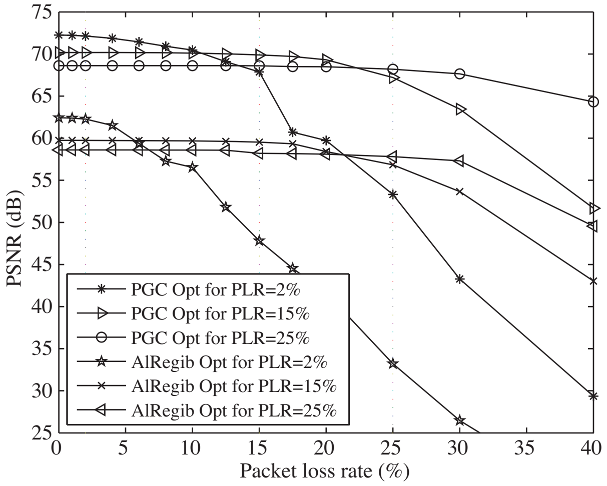


Figure 12. *Bunny* model is coded at 3.5 bpv and FEC assignment is optimized with respect to three different $P_{\rm LR}$'s for *PGCStankovic* and *Al-Regib kStep* = 5 methods. Performance of the three different assignments for various $P_{\rm LR}$'s in terms of simulated distortion in PSNR scale.

The first observation in the figure is that, when the transmission packetization is optimized for a low loss rate and a channel with a higher loss rate is encountered, the performance degradation can be severe. On the other hand, when the model encounters a channel with a lower loss rate, the performance loss is not significant. Another observation is that both CPM and PGC based methods behave similarly in mismatch scenario and the performance gap between the methods does not vary.

5.6. Complexity comparison

In the packet loss resilient 3D mesh transmission frameworks presented in this paper, it should be noted that the methods are not suitable for real time transmission. The reason is that the algorithms need to compress the model first and obtain D-R curves, which require significant amount of time. However, if we are allowed to compress models and store D-R curves offline, then real time transmission may be possible unless the time spent during the optimization, so called optimization time, is high. In this case, the optimization time of an algorithm becomes an important measure for real time transmission.

In order to compare the complexities of the optimization parts of the methods, we measured the optimization time for each algorithm as described at the beginning of Section 5. Table 4 summarizes time requirements for different optimization schemes mentioned in this paper.

The table shows that *PGCStankovic* has the smallest complexity. However, as seen in previous results, the complexity is smaller than that of *PGCMohr* at the cost of occasional slightly worse simulated PSNR. Examining the CPM based methods, we see that *Ahmad* and *Al-Regib kStep=5* show significantly higher

Table 4. Optimization times of different methods. Each method is optimized for $P_{\mathrm{LR}} = 4\%$.

Optimization Times		
Methods	Time (sec)	PSNR
PGCMohr	0.250	71.63
PGCStankovic	0.004	71.68
Ahmad	11.110	61.37
Al-Regib kStep = 5	8.350	61.36
Al-Regib kStep = 8	0.365	61.35

complexity. However, from $kStep = 8$, the complexity values are comparable with those of PGC based methods.

5.7. Visual comparison of CPM and PGC mased methods

Apart from the objective distortion metric results, we also present results of *Bunny model coded at 3.5 bpv* and *Venus head model coded at 4 bpv* in terms of visual reconstructions. Figure 13 and Figure 14 show visual reconstructions for *Bunny* and *Venus head* model for various P_{LR} values respectively.

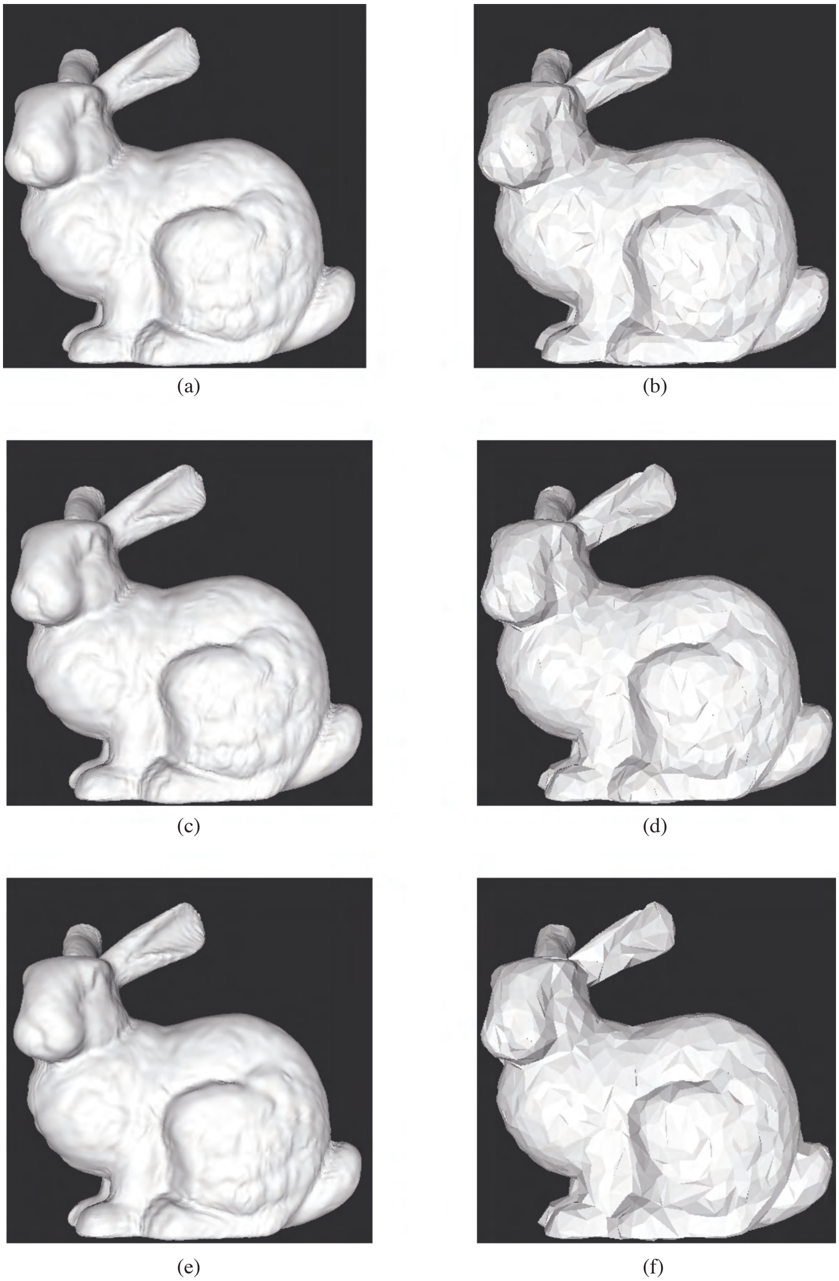
6. Conclusion

In this paper, we presented an extensive analysis of loss resilient coding methods for 3D models. The methods are based on optimally protecting compressed bitstreams with FEC codes with respect to given channel bandwidth and packet loss rate constraints.

We first examined the CPM based methods reported in the literature and came up with a general problem definition and solution. We introduced a *kStep* parameter to iterate protection rates with different steps and showed that increasing *kStep* considerably decreases optimization times at the expense of very small PSNR degradation.

Then we compared CPM and PGC based methods and experimental results show that PGC methods achieve approximately 10 db better PSNR for all loss rates. It was already reported in [12] that, compression performance of PGC method is 10 dB better than CPM method. In our results, we show that the 10 dB compression performance gap between the methods is preserved in packet loss resilient transmission. For the same reason, expected reconstructed models of PGC method at the decoder have a better subjective quality than the ones of CPM method. Apart from the PSNR performance, PGC based methods have an advantage of flexible packetization. Since the bitstream of PGC method is embedded, the bitstream is generated only once. Given the channel and bandwidth conditions, the bitstream can be truncated to desired bitrate precisely and FEC assignment is performed easily. The CPM based methods need to generate different bitstreams for different quantization values and number of bitrate values that can be achieved is limited.

Finally, we simulated performance of optimization methods in a mismatch scenario i.e. the model is protected with FEC codes optimized for a given loss rate but transmitted through a channel with a different loss rate. We observed that when the model is optimized for a low loss rate and encounters a channel with a higher loss rate, the performance degradation can be severe. On the other hand, when the model encounters a channel with a lower loss rate, the loss in the performance is not significant. Therefore we conclude that when the channel conditions are uncertain or time varying, it is more robust to optimize loss protection with respect to a higher loss rate.

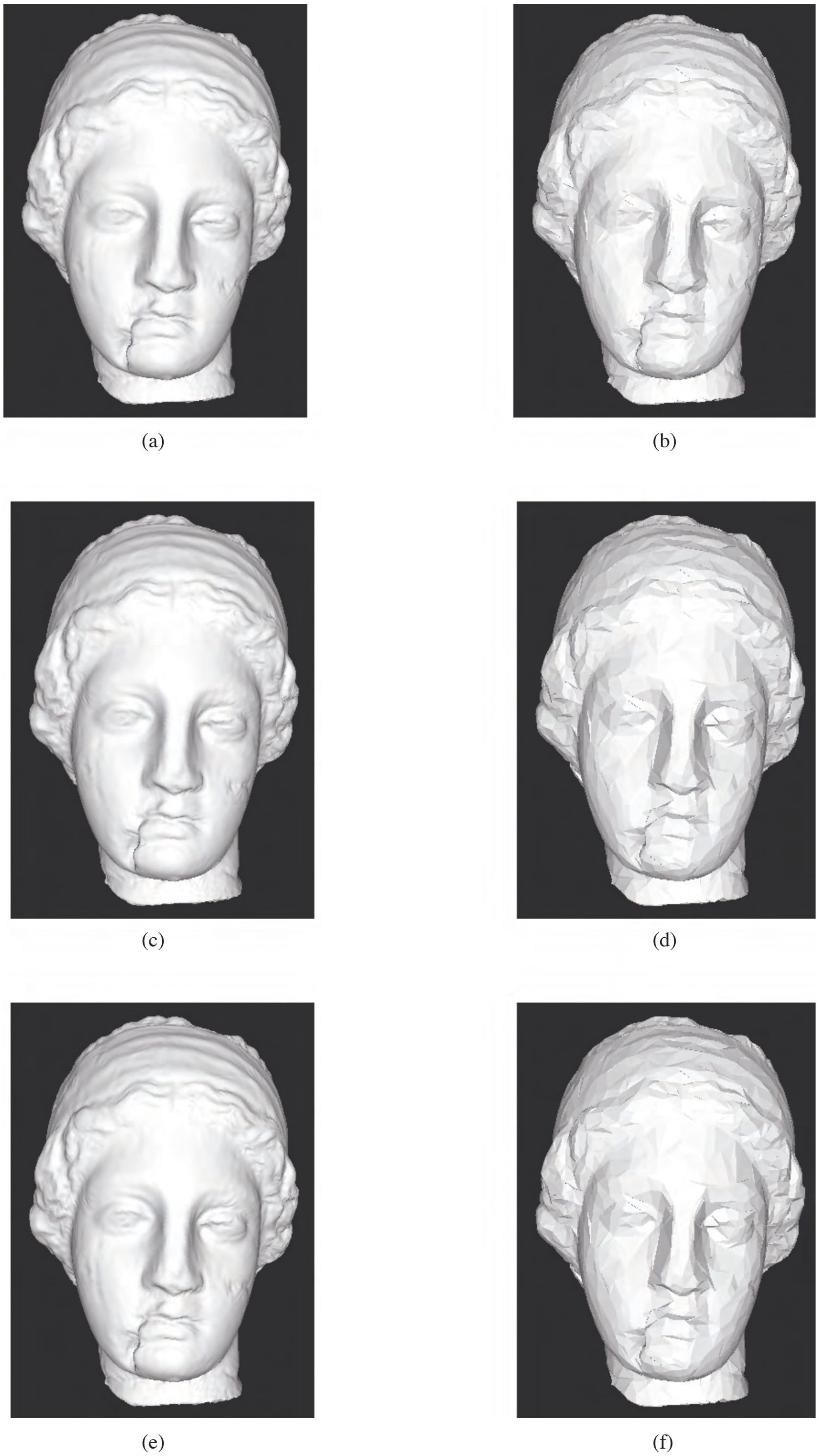


(a) (b)

(c) (d)

(e) (f)

Figure 13. Expected reconstructions of the *Bunny* model, left column by *PGCStankovic* method and right column by *Al-Regib kStep = 5* method. Images (a)–(b): $P_{\mathrm{LR}} = 2\%$; (c)–(d): $P_{\mathrm{LR}} = 10\%$; (e)–(f): $P_{\mathrm{LR}} = 20\%$. PSNR values: (a) 72.18 dB (b) 62.44 dB (c) 70.52 dB (d) 61.57 dB (e) 69.09 dB (f) 58.61 dB.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 14. Expected reconstructions of the *Venus head* model, left column by *PGCStankovic* method and right column by *Al-Regib kStep = 5* method. Images (a)–(b): $P_{\mathrm{LR}} = 2\%$; (c)–(d): $P_{\mathrm{LR}} = 10\%$; (e)–(f): $P_{\mathrm{LR}} = 20\%$. PSNR values: (a) 76.95 dB (b) 67.79 dB (c) 75.04 dB (d) 65.19 dB (e) 74.24 dB (f) 65.19 dB.

One of the future works includes extension of the error resilient techniques to the recently popular area of dynamic meshes which compose of series of static meshes. Another future work includes comparing with other scalable mesh coding techniques such as [40].

Acknowledgment

We would like to thank Andrei Khodakovsky for PGC software, Alexander Mohr for making his bit allocation algorithm publicly available, Ghassan Al-Regib, Vladimir Stankovic and Shakeel Ahmad for helps in generating results of their algorithms. Bunny model is courtesy of Stanford. This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV. It is also partially supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK).

References

[1] P. Alliez and C. Gotsman, "Recent advances in compression of 3D meshes," in *Proc. Symposium on Multiresolution in Geometric Modeling*, 2003.

[2] J. Peng, C.-S. Kim, and C.-C. J. Kuo, "Technologies for 3D mesh compression: A survey," *Journal of Visual Communication and Image Representation*, vol. 16, pp. 688–733, Dec. 2005.

[3] C. L. Bajaj, S. Cutchin, V. Pascucci, and G. Zhuang, "Error resilient transmission of compressed vrml," *TICAM, The Univ. Texas at Austin, Austin, TX, Tech. Rep., 1998.*

[4] 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, pp. 860–873, Jul. 2001.

[5] I. Cheng and A. Basu, "Perceptually optimized 3D transmission over wireless networks," in *International Conference on Computer Graphics and Interactive Techniques.* ACM New York, NY, USA, 2005.

[6] I. Cheng, L. Ying, K. Daniilidis, and A. Basu, "Robust and scalable transmission of arbitrary 3D models over wireless networks," *Journal on Image and Video Processing*, vol. 2008, no. 4, 2008.

[7] B. Yang, F. Li, Z. Pan, and X. Wang, "An Effective Error Resilient Packetization Scheme for Progressive Mesh Transmission over Unreliable Networks," *Journal of Computer Science and Technology*, vol. 23, no. 6, pp. 1015–1025, 2008.

[8] P. Jaromersky, X. Wu, Y. Chiang, and N. Memon, "Multiple-description geometry compression for networked interactive 3D graphics," in *Proc. ICIG'2004*, Dec. 2004, pp. 468–471.

[9] M. O. Bici and G. Bozdagi Akar, "Multiple description scalar quantization based 3D mesh coding," in *Proc. IEEE Int. Conf. Image Processing*, Atlanta, US, Oct. 2006, pp. 553–556.

[10] A. Norkin, M. O. Bici, G. B. Akar, A. Gotchev, and J. Astola, "Wavelet-based multiple description coding of 3-D geometry," in *VCIP'07, Proc. SPIE*, vol. 6508, San-Jose, US, Jan. 2007, pp. 65 082I–1 –65 082I–10.

[11] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, June 1996.

[12] A. Khodakovsky, P. Schröder, and W. Sweldens, “Progressive geometry compression,” in *Siggraph 2000, Computer Graphics Proceedings*, 2000, pp. 271–278.

[13] G. AlRegib, Y. Altunbasak, and J. Rossignac, “An unequal error protection method for progressively transmitted 3-D models,” *IEEE Trans. Multimedia*, vol. 7, pp. 766–776, Aug. 2005.

[14] G. AlRegib, Y. Altunbasak, and R. M. Mersereau, “Bit allocation for joint source and channel coding of progressively compressed 3-D models,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 256–268, Feb. 2005.

[15] G. AlRegib, Y. Altunbasak, and J. Rossignac, “Error-resilient transmission of 3-D models,” *ACM Trans. on Graphics*, vol. 24(2), pp. 182–208, Apr. 2005.

[16] S. Ahmad and R. Hamzaoui, “Optimal error protection of progressively compressed 3D meshes,” in *Proceedings IEEE International Conference on Multimedia and Expo (ICME’06)*, Jul 2006, pp. 201–204.

[17] M. O. Bici, A. Norkin, and G. Bozdagi Akar, “Packet loss resilient transmission of 3D models,” in *Proc. IEEE Int. Conf. Image Processing (ICIP 2007) to be presented*, San Antonio, US, 2007. [Online]. Available: http://www.eee.metu.edu.tr/˜mobici

[18] R. Pajarola and J. Rossignac, “Compressed progressive meshes,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 79–93, Jan. 2000.

[19] A. E. Mohr, E. A. Riskin, and R. E. Ladner, “Graceful degradation over packet erasure channels through forward error correction,” *Proceedings of the 1999 Data Compression Conference (DCC)*, 1999.

[20] S. Dumitrescu and X. Wu, “Globally optimal uneven erasure-protected multi-group packetization of scalable codes,” in *Proceedings IEEE International Conference on Multimedia and Expo (ICME’05)*, Jul 2005, pp. 900–903.

[21] A. Khodakovsky and I. Guskov, “Compression of normal meshes,” in *Geometric Modeling for Scientific Visualization. Springer-Verlag*, 2003.

[22] F. Morán and N. García, “Comparison of wavelet-based three-dimensional model coding techniques,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 7, pp. 937–949, July 2004.

[23] S. Lavu, H. Choi, and R. Baraniuk, “Estimation-quantization geometry coding using normal meshes,” in *DCC ’03: Proceedings of the Conference on Data Compression.* Washington, DC, USA: IEEE Computer Society, 2003, p. 362.

[24] F. Payan and M. Antonini, “3D mesh wavelet coding using efficient model-based bit allocation,” *3dpvt*, vol. 00, p. 391, 2002.

[25] F. Morán, P. Gioia, M. Steliaros, M. Bourges-Sévenier, and N. García, “Subdivision surfaces in MPEG-4,” in *Proc. IEEE ICIP*, vol. III, Sept 2002, pp. 5–8.

[26] M. Bourges-Sévenier and E. S. Jang, “An introduction to the MPEG-4 animation framework eXtension,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 7, pp. 928–936, July 2004.

[27] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, “MAPS: Multiresolution adaptive parameterization of surfaces,” *Computer Graphics*, vol. 32, no. Annual Conference Series, pp. 95–104, 1998.

[28] C. Loop, "Smooth subdivision surfaces based on triangles," *Masters thesis, Univ. of Utah, Dept. of Mathematics*, 1987.

[29] N. Dyn, D. Levin, and J. A. Gregory, "A butterfly subdivision scheme for surface interpolation with tension control," *ACM Trans. Graph.*, vol. 9, no. 2, pp. 160–169, Apr. 1990.

[30] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proc. Graphics Interface*, Vancouver, BC, Canada, Jun. 1998.

[31] A. Mohr, E. Riskin, and R. Ladner, "Approximately optimal assignment for unequal loss protection," in *Proc. ICIP'00*, 2000, pp. 367–370.

[32] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correctioncodes," in *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, vol. 1, Oct 1999.

[33] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," in *DCC '02: Proceedings of the Conference on Data Compression*, Apr. 2002.

[34] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Packet loss protection of embedded data with fast local search," *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 2, 2002.

[35] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, pp. 167–174, 1998.

[36] Y. Charfi, R. Hamzaoui, and D. Saupe, "Model-based real-time progressive transmission of images over noisy channel," in *Proc. WCNC'03*, New Orleans, LA, Mar. 2003, pp. 347–354.

[37] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *The Bell System Technical Journal*, vol. 39, no. 9, pp. 1253–1265, 1960.

[38] B. Girod, K. Stuhlmüller, M. Link, and U. Horn, "Packet loss resilient internet video streaming," *Proceedings of SPIE Visual Communications and Image Processing '99*, pp. 833–844, 1999.

[39] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Image Com.*, vol. 15, no. 1–29, pp. 77–94, Sep 1999.

[40] D. C. Cernea, A. , Munteanu, A. Alecu, A. Cornelis, and P. Schelkens, "Scalable Joint Source and Channel Coding of Meshes," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 503–513, 2008.