

Robust and Scalable Transmission of Arbitrary 3D Models over Wireless Networks¹

Irene Cheng^{2,3}, Lihang Ying³, Kostas Daniilidis² and Anup Basu³

²Department of Computer and Information Science, University of Pennsylvania, USA

³Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada

Email:chenglin@seas.upenn.edu, kostas@cis.upenn.edu, {lin, lihang, anup}@cs.ualberta.ca

Abstract

In this work we describe transmission of 3D objects represented by texture and mesh over unreliable networks, extending our earlier work for regular mesh structure to arbitrary meshes and considering linear vs. cubic interpolation. Our approach to arbitrary meshes considers stripification of the mesh and distributing nearby vertices into different packets, combined with a strategy that does not need texture or mesh packets to be re-transmitted. Only the valence (connectivity) packets need to be re-transmitted; however, storage of valence information requires only 10% space compared to vertices and even less compared to photo-realistic texture. Thus, less than 5% of the packets may need to be re-transmitted in the worst case to allow our algorithm to successfully reconstruct an acceptable object under severe packet loss. Even though packet loss during transmission has received limited research attention in the past, this topic is important for improving quality under lossy conditions created by shadowing and interference. Results showing the implementation of the proposed approach using linear, cubic and laplacian interpolation are described, and the mesh reconstruction strategy is compared with other methods.

I. INTRODUCTION

The benefit of designing packets optimizing texture-mesh tradeoff was discussed in our earlier work [9]. However, our previous research was restricted to regular meshes, limiting the application of the algorithms. In this work we extend earlier research by taking transmission of arbitrary meshes into account. To limit the scope of the current work, we only consider mesh transmission in this report. Detailed surveys on simplification algorithms can be found in [22, 30]. These algorithms try to control the complexity of a mesh and preserve surface structures by developing various strategies for generating Level-of-Detail (LoD) in different parts of a 3D object. An example of geometric simplification is shown in Figure 1, in which the Buddha model is simplified to various resolution levels (number of faces are 3,000 left, 1,000 middle and 500 right). There exists substantial literature on multimedia transmission over wireless networks, such as [53, 54]. However, much less research has addressed wireless 3D transmission. The importance of 3D wireless transmission has grown with the advent of the IEEE 802.11 card on most laptops, the popularity of 3D online games on handheld devices, and the emerging 3D TV marketplace [8, 17]. In [2, 11] robust wireless transmission of mesh over wireless networks has been discussed. However, these methods do not take joint texture and mesh transmission into account. In addition, the proposed algorithms assume that some parts of the mesh can be transmitted without loss over a wireless network, allowing progressive mesh transmission to give good results. The limitation of this assumption is that application layer protocols must be deployed [1], and some re-transmission may be necessary. Also, some of the approaches proposed earlier assume bit error correction rather

¹ Parts of this work have been presented at the IEEE Symposium on 3D PVT, Chapel Hill, USA, June 2006. Irene Cheng was also a visiting fellow at the Department of Computer and Information Science, University of Pennsylvania. The support of Alberta Informatics Circle of Research Excellence (iCORE) and NSERC in making this work possible is gratefully acknowledged.

than lost packets. Packet loss probability models have been proposed by some researchers, *e.g.* [28]; however, these models are usually associated with retransmission. In order to make our algorithms work over an arbitrary wireless environment, we simply assume packet-based transmission where a certain percentage of the packets may be lost. The approach proposed in [9] assumed a regular mesh, thus creating packets was fairly straightforward. In this work, we propose a strategy to packetize arbitrary meshes to reduce the effect of loss during transmission.

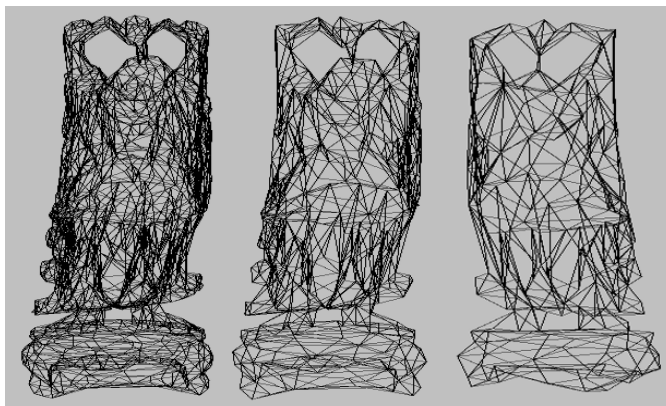


Figure 1: Buddha model at various mesh resolution levels.

Even though most papers do not consider packet loss rates beyond 10% for wired networks, we consider higher loss rates considering “shadowing” and interference in wireless networks, which could be ad hoc [58] (where hosts depend on one another to keep the network connected) and follow peer-to-peer transmission strategies as well.

With the demand on tetherless connectivity, there has been a surge of research activities in the area of wireless communication [50]. Differing from wired communication, wireless communication has two challenging aspects: First, is the fading phenomenon, which includes small-scale multipath fading and larger-scale fading such as path loss via distance attenuation and shadowing by obstacles. Second is interference, which could be between transmitters

communicating with a common receiver, between multiple receivers communicating with a single transmitter, or between different transmitter-receiver pairs. These lossy conditions are often encountered when entering a basement of a building, driving under a bridge, or when many users try to get onto a wireless network in a hotel lobby.

There is significant research on packet loss in wireless network. The authors in [52] conducted sensor node's field test to measure packet loss rate against distance and transmission power. The tests observed that packet loss rate increases up to 100% by increasing the distance and decreasing the transmission power. Others [15, 39] studied packet loss due to interference between IEEE 802.11b and Bluetooth devices. In the presence of IEEE 802.11b interference with strong signal strength, the percentage of lost UDP packets in Bluetooth transmission could be 70%. Mazzenga and others [33, 34] describe the packet loss probability in an environment with many piconets. (A piconet is an ad hoc network of devices connected by Bluetooth.) With 40 piconets in an area of $20 \times 20 \text{ m}^2$, the packet loss probability could be up to 60%.

The authors in [2] consider packet loss up to 40%. In [11], partial data is transmitted by UDP and the work considers the situation of receiving 300,000 faces out of 1.08M faces, which is equivalent to more than 70% packet loss. In the context of multi-description transmission [25], only 1 out of 4 descriptions is considered to be transmitted due to limited bandwidth. In multicast or broadcast situation, no acknowledgement and retransmission is possible. When the bandwidth of one specific client is fluctuating, the amount of data received could vary. Several papers discuss novel strategies for wireless network management, including QoS provisioning, hybrid channel allocation, and database and location management schemes for wireless networks [5, 19, 20, 21,

23]; however, the present paper will not address the possibility of optimizing our algorithms considering these advanced wireless network management protocols.

Our proposed approach has two main components: distribution of neighboring vertices into different packets; and evaluation of alternative strategies for 3D interpolation based on surface reconstruction error. The issue of texture-mesh tradeoff optimizing perceptual quality [6, 7, 32, 36], described in detail in [9, 10], will not be discussed in this work; extensions relating to this area will be considered in the future.

The remainder of this paper is organized as follows: Section 2 reviews 3D mesh coding for transmission. Section 3 describes transmission strategies for irregular meshes. Experimental results on irregular mesh transmission under packet loss are described in Section 4. Section 5 compares the effectiveness of alternative interpolation strategies in reconstructing meshes recovered after packet loss. The effect of packetization on mesh compression is discussed in Section 6. Finally, Section 7 gives the conclusions and discusses future work.

II. 3D MESH CODING FOR TRANSMISSION

A 3D mesh is represented by geometry and connectivity [45]. An uncompressed representation, such as the VRML ASCII format [50], is inefficient for transmission. 3D mesh compression schemes usually handle geometry data following three steps: quantization, prediction, and statistical coding. However, algorithms differ from one another with respect to connectivity compression.

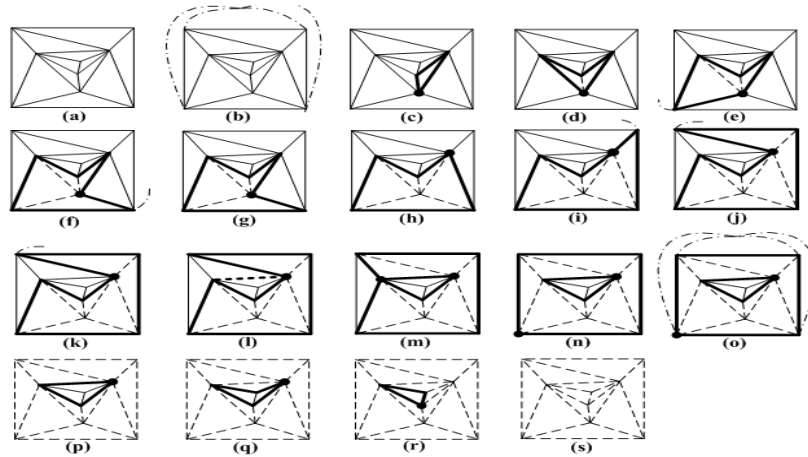


Figure 2: An example of a run of the valence-driven connectivity encoding algorithm. The active lists are indicated by thick lines, and edges already visited (encoded) by dashed lines.

Among the many 3D mesh compression schemes proposed since the early 1990s [38], the valence-driven approach [47] is considered to be a state-of-the-art technique [3, 4] for 3D mesh compression, with a compression rate of 1.5 bits per vertex on the average to encode mesh connectivity. However, this approach is restricted to manifolds [38]. A number of 3D mesh compression algorithms have been accepted as international standards. For example, Topological Surgery [49] and Progressive Forest Split [48] have been adopted by VRML version 2 [51] and MPEG-4 version 2, defined as 3D Mesh Coding (3DMC) [35].

The valence-driven algorithm begins by randomly selecting a triangle. Starting from a vertex of that triangle and traversing all the edges in a counter-clockwise direction (Figure 2), the visited vertices are pushed into an *active list*. After visiting the associated edges, the next vertex is popped from the *active list*, and the process is repeated. The valence (or degree) of each processed vertex is output. From the stream of vertex valences, the original connectivity can be reconstructed, as shown in Figure 3.

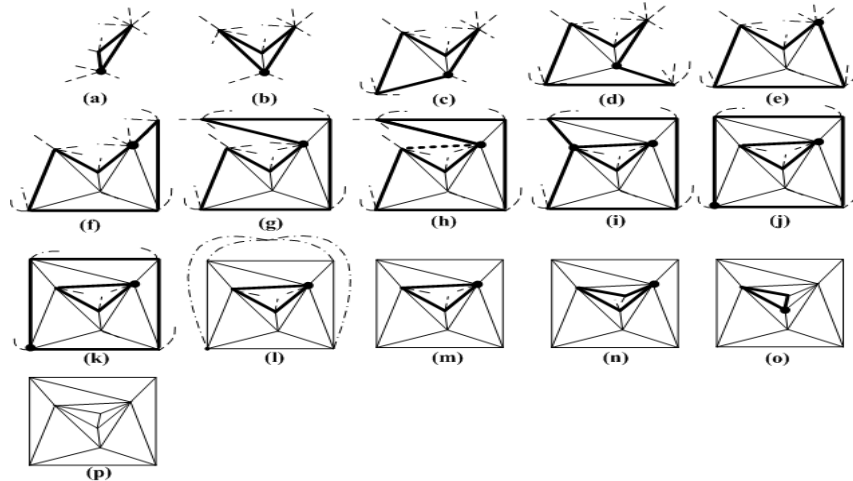


Figure 3: An example of connectivity decoding (or reconstructing) from the stream of vertex valences in the valence-driven algorithm.

There are many other innovative approaches for mesh and connectivity coding and compression, including topological surgery [49], Progressive Forest Split [48], MPEG-4 3D Mesh Coding (3DMC) [37], and so on. A detailed review of these papers can be found in [10], and is thus not included here.

Current 3D mesh coding techniques mainly focus on coding efficiency, *i.e.* compression ratio, by transmitting incremental data. This approach is good without packet loss but is vulnerable to channel errors for irregular meshes. Figure 4 shows an example of error sensitivity of the Edgebreaker 3D mesh coding method [29, 43]. With one error character in the connectivity stream, the decoded mesh can change significantly and can be impossible to reconstruct.

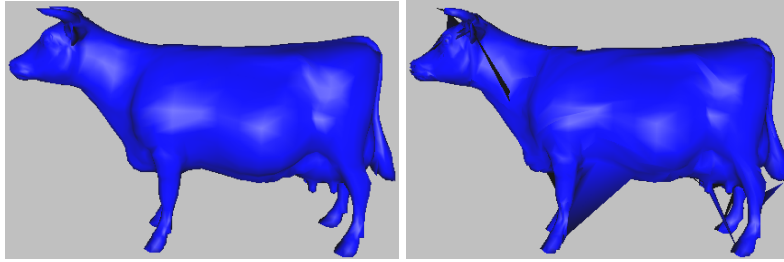


Figure 4: An example of error sensitivity of the Edgebreaker 3D mesh coding method. **Left:** original 3D mesh; **Right:** Decoded 3D mesh with one error character in the decoded connectivity stream.

To transmit compressed 3D meshes over a lossy network, there are two approaches. The first approach is to compress 3D meshes in an error-resilient way. [57] proposed partitioning a mesh into pieces with joint boundaries and encoding each piece independently. However, if packets are lost, there are holes in the mesh resulting from missing pieces. [25] introduced multiple description coding for 3D meshes. Each description can be independently decoded. But it assumes that the connectivity data is guaranteed to be correctly received. The second approach is to use error protection to restore lost packets [2, 42].

Instead of transmitting duplicate packets to reduce the effect of packet loss, we adopt a perceptually optimized statistical approach in which adjacent vertices and connectivity information are transmitted in different packets so that the possibility of losing a contiguous segment of data is minimized. Furthermore, our model takes both geometry and texture data into consideration, while previous approaches discuss only geometry. In the next section, we will discuss how our prior approach for joint texture-mesh transmission of regular meshes can be extended to work with irregular meshes.

III. TRANSMISSION STRATEGY FOR IRREGULAR MESHES

In prior work we discussed how adjacent vertex information could be distributed over separate packets so that the reconstructed 3D object can maintain satisfactory visual quality considering packet loss. However, in the experiments we assumed a regular or semi-regular mesh where connectivity information can easily be interpolated without significant loss of quality. Also, interleaving the original regular mesh data into packets was fairly straightforward by simply selecting vertices at pre-determined steps along two directions starting from a given vertex. In this section we will extend our transmission strategy over unreliable networks to irregular meshes. We will also analyze the performance of various 3D mesh interpolation strategies when only partial information is received at a client site.

When transmitting irregular mesh data, not only vertex information but also connectivity information plays a crucial role in 3D reconstruction at the client site. In order to preserve the original geometry of the object, many transmission algorithms suggest retransmission [12] of the base layers to safeguard the successful transmission of important features of the object [2, 31]. In progression-based algorithms [3, 24, 45, 46, 48, 56], the base layer cannot be lost. Retransmission adds an overhead on bandwidth limited connections, in particular on wireless and mobile networks. Without the need to retransmit the base layer, our goal is to find a trade-off between compression rate and robustness to packet loss. For example, although the Edgebreak 3D mesh coding method discussed in Section 2 has high compression ratio, the cow object (Figure 4) shows significant distortion even when one character in the connectivity chain is lost. In our strategy, we focus on the following criteria:

1. ***Efficient compression based on stripification*** – In order to avoid the memory bus bandwidth bottleneck in the processor-to-graphics pipeline and maintain high compression ratio, compression algorithms often employ a “tristrips” encoding method, which virtually specifies a triangulation cost of one vertex per triangle [16, 55] instead of sending three vertices per triangle. Figure 5 shows an example of applying stripification to a cow mesh. High compression ratio can be achieved if a mesh can be broken down into a few long continuous strips. In our approach, we traverse the vertices following the valence-driven method discussed in Section 2 because this algorithm generates long continuous tristrips.



Figure 5: Applying stripification to a cow mesh [55]. Different colors represent different triangle strips. (<http://www.cosy.sbg.ac.at/~held/projects/strips/strips.html>)

2. ***Robustness to packet loss based on distribution of neighboring vertices into different packets***
– In addition to stripification, we need to distribute neighboring vertex and connectivity information into different packets to minimize the risk of lost data affecting a large neighborhood. Let the total number of packets transmitted be p . Starting from the first vertex, traverse the vertices as in the valence driven approach. The first p vertices are distributed to p different packets. The process is repeated with the next p vertices, and so on. In other words, the possibility of lost adjacent vertices creating a large void region is reduced. The valence

information, which has a size of roughly 10% of the vertex information, is transmitted separately without loss; i.e., if packet(s) containing valence information are lost they are re-transmitted.

3. ***Texture-mesh tradeoff based on perceptual optimization*** – This topic will be an extension of our previous work [9, 10], and will be considered in future work.

Encoding Order and Packet Grouping

The encoding order and packet grouping can be explained by the color-coding scheme in Figure 6. Vertices with the same color are included in the same group. For example, the red colored vertices are grouped into the first packet; the lime colored ones put in the second packet, and so on. Figure 6, from left to right, shows the grouping of 32 vertices when 2, 4, 8 and 16 packets are used.

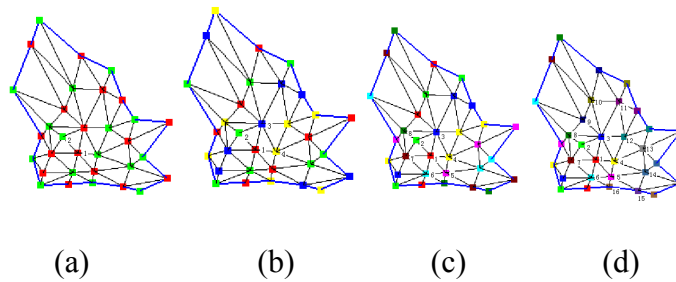


Figure 6: (a) 2 packets; (b) 4 packets; (c) 8 packets; (d) 16 packets.

Interpolation of Lost Geometry

After all the packets are received, first, the mesh is partially reconstructed based on the geometry packets received and connectivity; following the same order as in the encoding process. Then, the

vertices are traversed in the reconstruction order of the valence-driven decoding algorithm. When a vertex with lost geometry, L , is encountered, the adjacent reconstructed vertices with an edge connected to L , whose geometry is either not lost or is interpolated previously, are used to interpolate the geometry of L . Several interpolation strategies, linear, cubic and laplacian were considered. Brief pseudo-code of an interpolation method is given in the Appendix.

IV. EXPERIMENTAL RESULTS FOR IRREGULAR MESHES

In Figure 7, 0%, 30%, 50%, 60% and 80% randomly selected vertices were lost for a Cow mesh. However, the lost geometry was interpolated based on neighboring vertices and valence information, which is transmitted without error. It can be seen that smoothness on the object surface begins to deteriorate at about 60% loss. Visual degradation becomes more obvious at 80% loss; still, the object is recognizable as a cow.

Assuming 1.5 bits/vertex on the average to encode mesh connectivity [38], 13.3 bits/vertex to encode geometry [44], and 650 vertices and 50 Kbytes or higher for the compressed photorealistic textures in Figure 9, the cost of re-transmission of the connectivity information for this real example is less than 1%. Thus, to avoid the delays in requesting re-transmission of packets, it may be wiser to send duplicate packets containing the connectivity information so that real-time visualization of photo-realistic texture mapped 3D objects at high packet loss can be facilitated.

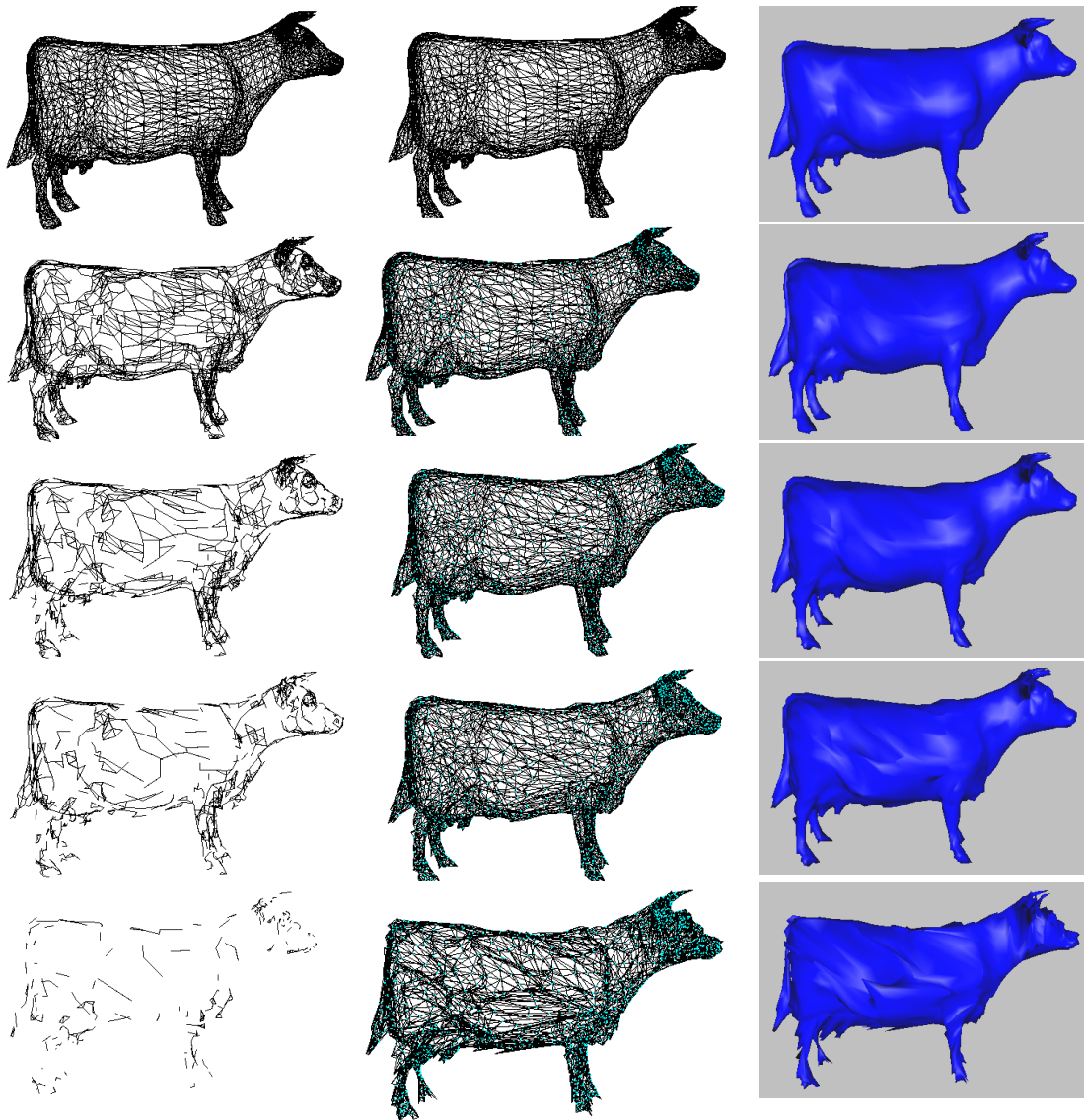


Figure 7: From top to bottom, (Left): 0%, 30%, 50%, 60% and 80% randomly selected packet loss was applied to a Cow mesh; (Middle): Interpolated meshes; (Right): The corresponding mesh mapped with color.

Next, we consider the effect of varying number of packets on data loss. Figure 8 shows how the vertices are assigned to 2, 4, 8 and 16 packets; with each color belonging to a specific packet.

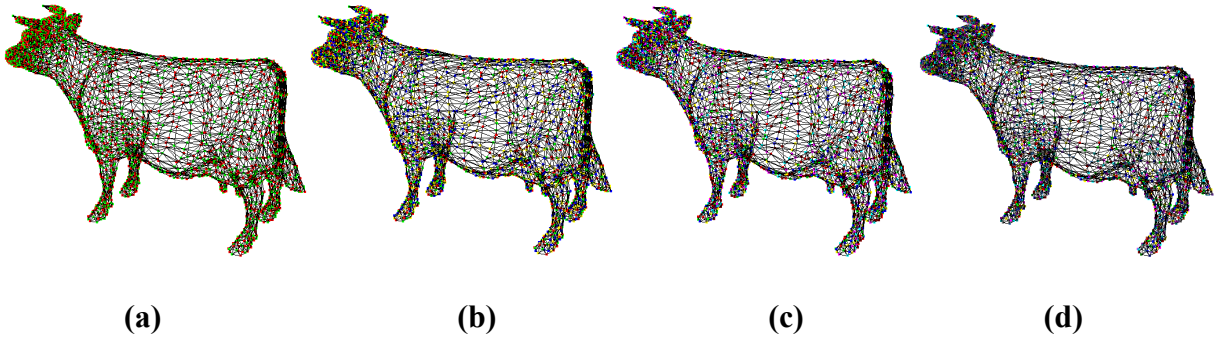


Figure 8: Cow vertices encoded in (a) 2 packets; (b) 4 packets; (c) 8 packets; (d) 16 packets.

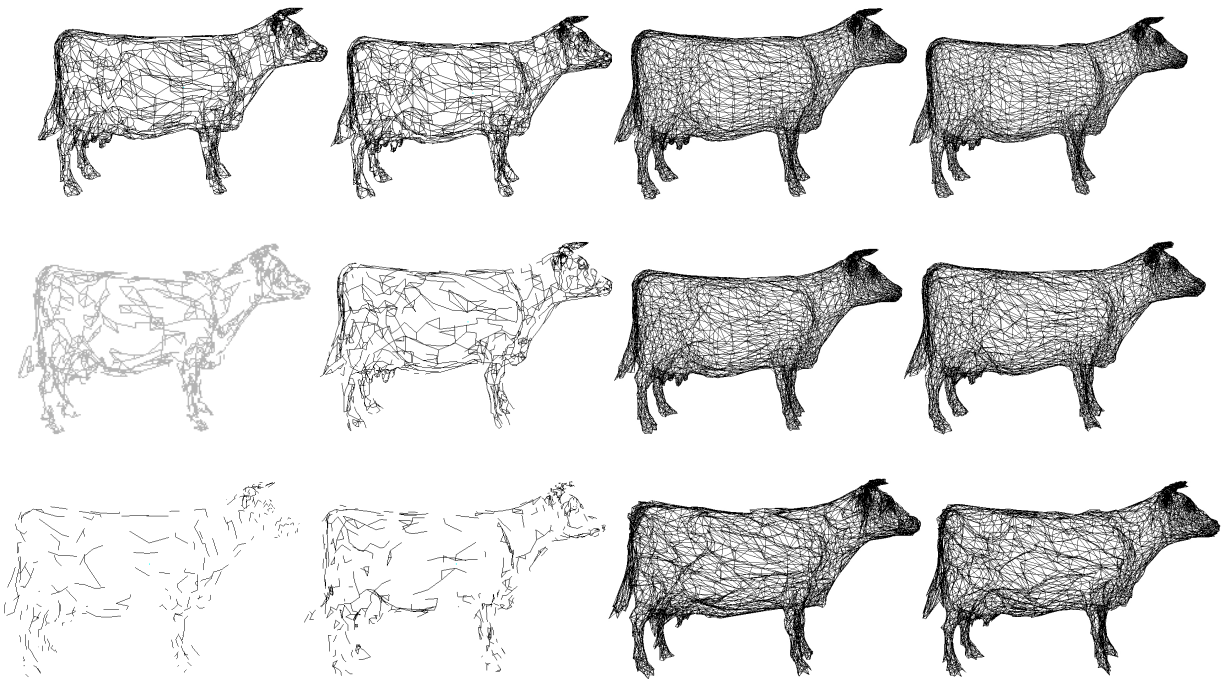


Figure 9: From top to bottom: **(Column 1, before interpolation):** 4 out of 16 packets lost; 8 out of 16 packets lost; 12 out of 16 packets lost; **(Column 2, before interpolation):** 1 out of 4 packets lost; 2 out of 4 packets lost; and 3 out of 4 packets lost; **(Column 3, after interpolation):** 4 out of 16 packets lost; 8 out of 16 packets lost; 12 out of 16 packets lost; **(Column 4, after interpolation):** 1 out of 4 packets lost; 2 out of 4 packets lost; and 3 out of 4 packets lost.

Figure 9 shows that the proportion of packets lost is more important than the number of packets used. Thus, the reconstructed meshes appear similar, regardless of whether 12 out of 16 or 3 out of 4 packets are lost. Figure 10 shows the results of our approach applied to other models for various packet loss rates.

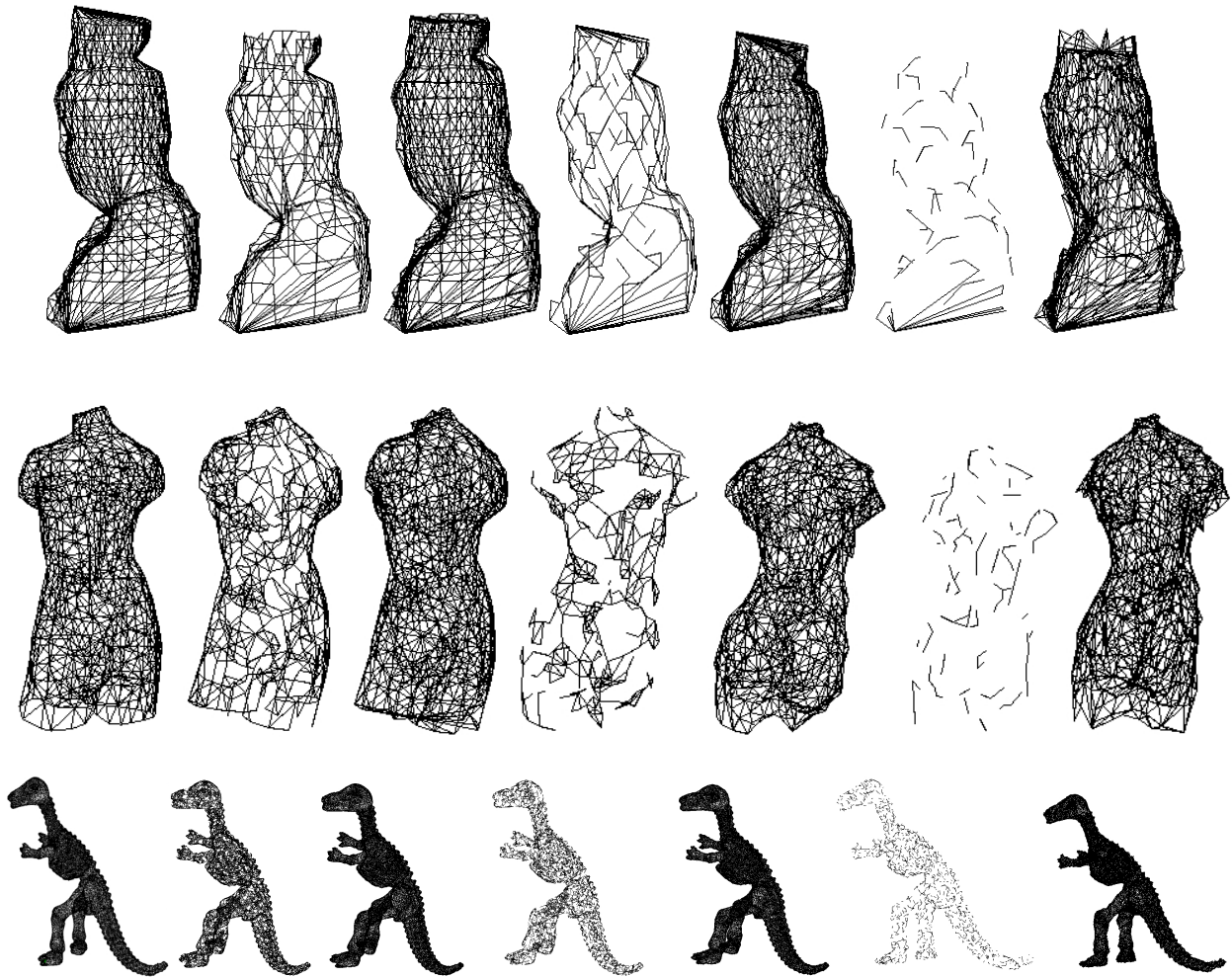


Figure 10: Different Models: queen (1st row, 650 vertices); body (2nd row, 711 vertices); dinosaur (3rd row, 14070 vertices). 1st column: original model; 2nd column: 4 loss out of 16 packets (before interpolation); 3rd column: 4 loss out of 16 packets (after interpolation); 4th column: 8 loss out of 16 packets (before interpolation); 5th column: 8 loss out of 16 packets (after interpolation); 6th column: 12 loss out of 16 packets (before interpolation); 7th column: 12 loss out of 16 packets (after interpolation).

To demonstrate the benefit of distributing nearby vertices into different packets we conducted experiments with packets containing nearby vertices. In this case, even the loss of 1 out of 16 packets can cause unacceptable distortions in the shape (Figure 11) compared to results obtained after much higher loss by our method (Figure 9).

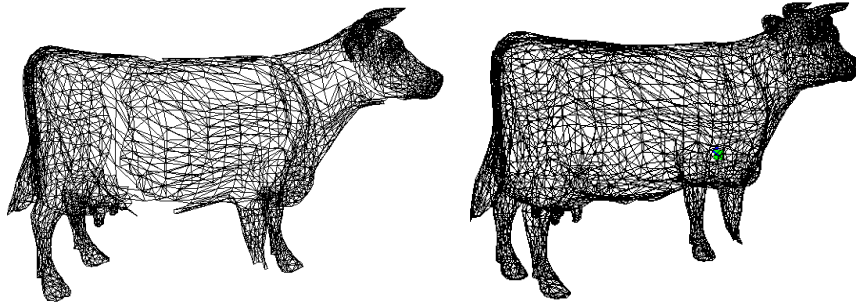


Figure 11: Effect of packet loss when nearby vertices are not distributed into different packets (1 out of 16 packets loss). Left: before interpolation; Right: after interpolation.

Some videos of our implementation results can be seen at: www.cs.ualberta.ca/~anup/SpecialIssue3D.

In the next section we compare some of the different approaches that can be used for interpolation of missing vertices.

V. Comparison of Different Interpolation Methods

We applied the triangle-based linear, triangle-based cubic spline and ‘v4’ [40] interpolation methods [18] with different neighbor levels on nine models. The nine models have different densities, with number of vertices varying from 428 to 5000. We considered different levels of packet loss as well. The numbers of lost packets (out of 16) in the experiments were 4, 8 and 12. We used the metro tool [14] to measure error between the original and reconstructed models following Hausdorff distance. The metro tool is based on surface sampling and *point-to-surface* distance computation. It samples vertices, edges and faces by taking a number of samples that is approximately 10 times the number of faces.

In Table I we can see that the triangle-based cubic spline interpolation method with neighborhood level equal to 1 (i.e., containing neighbors at distance 1 from a given vertex) has best overall performance — producing minimal reconstruction errors in most cases. The ‘v4’ method performs significantly poorer because the number of data points is not large enough and the slopes of the end data points are not constrained to be zero. Note that for several cases linear interpolation with neighborhood level of 1 outperforms the other approaches. The lowest error value in each row is marked with a “*” for all rows of the tables.

Number of Lost Packets (out of 16) = 4

Model(Vertex Number)	Reconstruction Error					
	Linear Interpolation		Cubic Interpolation		V4 Interpolation	
	Nhbr. Level =1	Nhbr. Level =2	Nhbr. Level =1	Nhbr. Level =2	Nhbr. Level =1	Nhbr. Level =2
Armadillo (1752)	9.04437(*)	10.68781	9.29891	10.31608	21523.9472	31.39821
Body(711)	0.293317	0.308802	0.282268(*)	0.293965	0.358016	0.289887
Bunny(2503)	0.003009(*)	0.003759	0.003010	0.003927	0.034500	0.010478
Cow(2904)	0.025903	0.032442	0.025100(*)	0.034776	0.058983	0.030875
Dinosaur(5000)	1.617305	1.703516	1.462922(*)	2.316251	628.801147	155.185516
Dragon(1252)	9.619162	9.619162	9.619162(*)	9.619162	1975.683105	20.720869
HammerHead(752)	0.025389	0.030961	0.025343(*)	0.031520	0.867992	0.701022
Mannequin(428)	0.274351(*)	0.368580	0.299820	0.405500	0.629864	0.463766
Queen(650)	0.112574	0.200955	0.111644(*)	0.187389	0.192037	1.974105

Number of Lost Packets (out of 16) = 8

	Reconstruction Error					
	Linear Interpolation		Cubic Interpolation		V4 Interpolation	
Model(Vertex Number)	Nhbr. Level =1	Nhbr. Level =2	Nhbr. Level =1	Nhbr. Level =2	Nhbr. Level =1	Nhbr. Level =2
Armadillo (1752)	14.01629	14.24620	14.01457(*)	14.21098	-	244.5687
Body (711)	0.324557	0.343627	0.323471	0.31086(*)	0.593218	0.326924
Bunny (2503)	0.004593(*)	0.005291	0.004615	0.005471	0.025934	0.033867
Cow (2904)	0.032304	0.034303	0.032176(*)	0.036055	0.082749	0.057212
Dinosaur (5000)	2.868300	3.228362	2.855550(*)	3.401511	-	99629.343
Dragon (1252)	15.491241	16.276470	15.459133(*)	16.276007	-	35.06437
HammerHead(752)	0.065599	0.070335	0.065599(*)	0.071985	0.293371	1.191587
Mannequin (428)	0.469657(*)	0.494803	0.478435	0.495934	0.710001	0.590717
Queen (650)	0.187299	0.226249	0.177390(*)	0.227999	0.278772	2.211618

Number of Lost Packets (out of 16) = 12

	Reconstruction Error					
	Linear Interpolation		Cubic Interpolation		V4 Interpolation	
Model(Vertex Number)	Nhbr. Level =1	Nhbr. Level =2	Nhbr. Level =1	Nhbr. Level =2	Nhbr. Level =1	Nhbr. Level =2
Armadillo (1752)	22.9973	15.5132(*)	23.0192	15.6067	-	5750.9697
Body (711)	0.6155	0.6494	0.6155(*)	0.6430	0.6984	1.5861
Bunny (2503)	0.008582	0.0105	0.0084(*)	0.0105	0.0464	0.0232
Cow (2904)	0.047938	0.052571	0.0477(*)	0.054300	0.153599	0.067901

Dinosaur (5000)	4.8435(*)	5.023284	4.888701	5.023284	-	-
Dragon (1252)	15.516(*)	17.569109	17.091991	17.569109	-	788.059631
HammerHead(752)	0.121758	0.1182(*)	0.122472	0.123254	1.093346	0.693335
Mannequin (428)	0.673878	0.776635	0.6707(*)	0.765230	0.896839	0.896235
Queen (650)	0.301478	0.269726	0.302202	0.26258(*)	0.279607	3.439980

Table I: Comparison of different interpolation methods. The numbers with (*) marked indicate minimum reconstruction error for a given model with the same number of lost packets. ('-' means a value larger than 100,000.)

Comparison with Other Approaches

One objective of this work is to reconstruct the surface of 3D meshes after transmission with packet loss and without retransmission. One approach in the literature reconstructs from oriented point sets [26] — for this method, only the coordinates and normals of points, without connectivity information, are transmitted. From the coordinates received and normals of points, the surface of 3D meshes could be reconstructed when some of the points are lost. One disadvantage of this approach is that the reconstructed meshes could form disjoint pieces if the points are sparse. Differing from this approach, our approach transmits connectivity information and can work well even on sparse meshes.

An alternative method is to reconstruct the surface from the partially received meshes by subdivision methods, such as Catmull-Clark subdivision method [13] and Sqrt(3) subdivision method [27]. The surface subdivision method is usually used to generate a denser and smoother

surface from a coarser surface. More than one vertex is added and their coordinates are interpolated during surface subdivision. In Catmull-Clark subdivision method [13], the coordinates of added vertices are interpolated following the cubic spline algorithm. Sqrt(3) subdivision method [27] differs from other subdivision methods by increasing the number of triangles in every step by a factor of 3 instead of 4.

	Reconstruction Error			
	# of Lost Packets (out of 16) = 1		# of Lost Packets (out of 16) = 2	
	Sqrt(3) Subdivision	Our Approach Cubic Interpolation Nhbr. Level =1	Sqrt(3) Subdivision	Our Approach Cubic Interpolation Nhbr. Level =2
Armadillo (1752)	6.529486	2.923688(*)	12.067476	3.829655(*)
Body(711)	0.251311□	0.115403(*)	0.513330	0.213674(*)
Bunny(2503)	0.003671	0.001699(*)	0.010211	0.002416(*)
Cow(2904)	0.028814	0.013196(*)	0.057889	0.013489(*)
Dinosaur (5000)	1.773165	0.770778(*)	3.733543	1.275865(*)
Dragon(1252)	8.214386	5.994485(*)	12.851923	8.348866(*)
HammerHead(752)	0.024065	0.010686(*)	0.060305	0.011494(*)
Mannequin(428)	0.328453	0.146919(*)	0.752031	0.259950(*)

Table II: Comparison with subdivision-based approach. The numbers with (*) marked are the minimal error in the reconstructed models for the same model with the same number of lost packets.

We compared the proposed approach with the subdivision-based approach. When packets are lost, the coordinates of partial vertices are lost, resulting in holes in the meshes. Before applying subdivision method to reconstruct the 3D meshes, we closed the holes with a new polygon by

connecting the boundaries of the holes. The added polygons were not planar if their vertices were not in a plane. If the coordinates of too many vertices were lost, holes in 3D meshes could not be closed. Therefore, the experiments were conducted only for two cases, when 1 or 2 packets out of 16 were lost. Table II shows the experiment results comparing Sqrt(3) subdivision-based approach with one step subdivision and the proposed approach. From the table, we can see that the proposed approach has significantly lower reconstruction errors for all cases. We also observed that Catmull-Clark subdivision-based method and Sqrt(3) subdivision-based method had similar performance, and the reconstruction error did not decrease significantly by using more subdivision steps (Table III).

Subdivision Method	Subdivision Step	Reconstruction Error
Sqrt(3) Subdivision	1	0.028814
	2	0.028742
	3	0.028725
	4	0.028646
Catmull-Clark Subdivision	1	0.028683

Table III: Comparison among different subdivision methods and subdivision steps. The test model is Cow.

Sorkine et al. [41] proposed a transformation of 3D coordinates by using the Laplacian matrix of the mesh in order to enable aggressive quantization without significant loss of visual quality. Their scheme does not take packet loss into account. To reconstruct 3D coordinates, a linear equation is solved using a least-squares solver. The problem with applying this method under packet loss is that losing the Laplacian values of a few points makes accurately solving the linear

equation impossible, resulting in significant reconstruction error. Figure 12 shows how the reconstructed Cow model (2904 vertices) can have significant distortions after losing 2% of the Laplacian values.

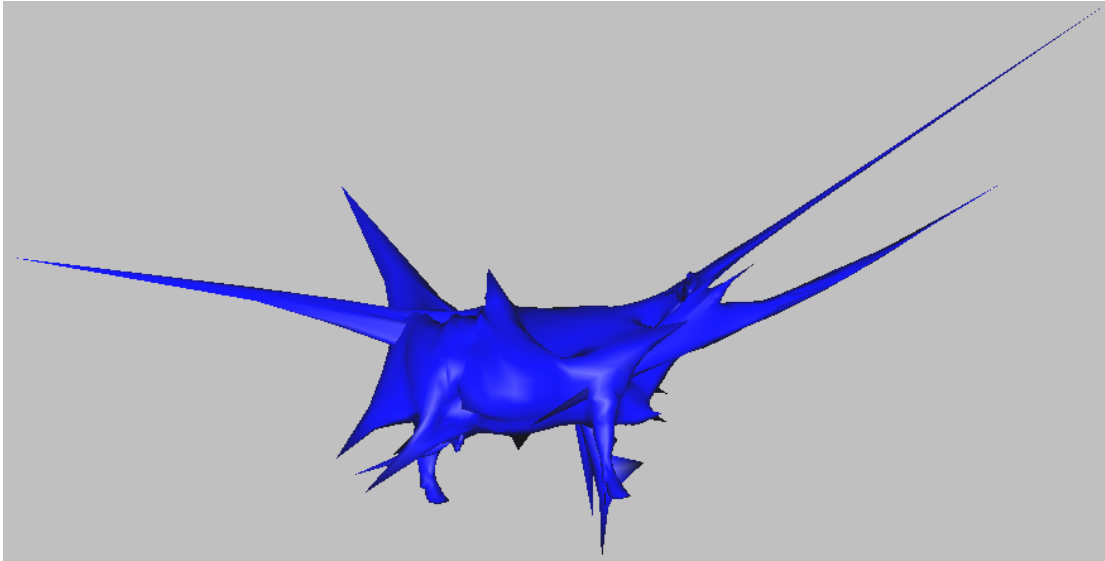
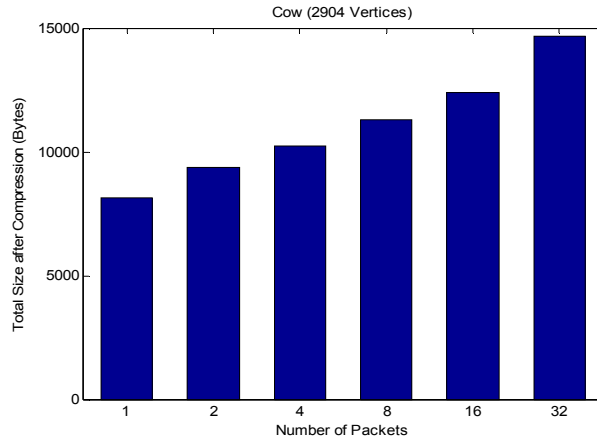


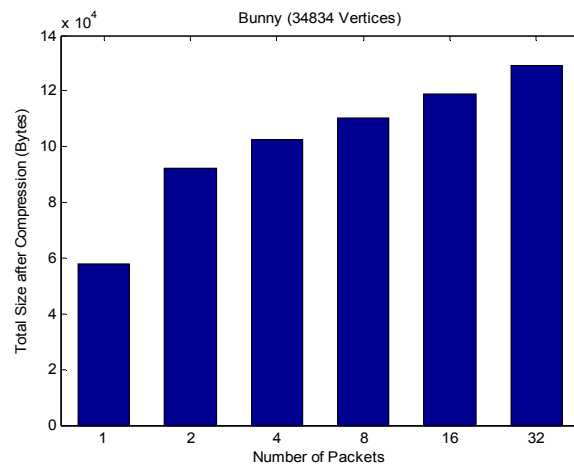
Figure12: The reconstructed Cow model (2904 vertices) after losing 2% of the Laplacian values.

VI. EFFECT OF PACKETIZATION ON MESH COMPRESSION

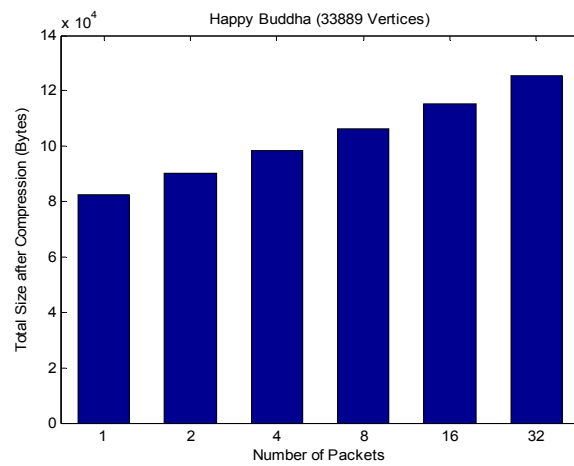
In order to support packet loss scenarios, in our scheme, each packet is compressed independently following the parallelogram predictive scheme. Thus, the loss of various packets does not affect each other. The compressed data in a received packet can be decompressed even if other packets are lost. The price of error resilient transmission is that the compression efficiency gets lower as the number of packets increases. Figure 13 shows the compression results relating to the total data size after compression considering different numbers of packets; 12-bit quantization was used in these experiments.



(a)



(b)



(c)

Figure 13: Total data size after compression considering different numbers of packets: (a) Cow (2904 Vertices); (b) Bunny (34834 Vertices); (c) Happy Buddha (33889 Vertices).

VII. CONCLUSION AND FUTURE WORK

In this paper we extended earlier work on 3D texture-mesh transmission with regular meshes to arbitrary meshes. Experimental results were shown with arbitrary meshes to demonstrate that the approach works well even when a high percentage of packets are lost.

We proposed a strategy of distributing neighboring vertex information into different packets to minimize the risk of lost data affecting a large neighborhood, and compared different interpolation schemes with different neighborhood levels during decoding. Experiments on models with different densities show that smoothness on the mesh surface deteriorates above 60% packet loss; but the objects we worked with are still recognizable. The reconstruction quality after transmission with packet loss depends on the original density of the model. Among the triangle-based linear spline, triangle-based cubic spline and ‘v4’ interpolation methods, the triangle-based cubic spline interpolation method performs best overall. Also, our reconstruction methods perform significantly better than subdivision-based approaches.

In future work we will consider developing robust strategies that work even when some of the connectivity information is lost.

Appendix A: Outline of An Interpolation Algorithm

A linear interpolation algorithm can be described as follows:

```
for each vertex  $v[i]$  in the reconstruction order of the valence-driven decoding algorithm {  
    if (the geometry of  $v[i]$  is lost) {  
        geometry_sum = 0; i = 0;  
        for each of the adjacent vertices  $a[j]$  with an edge connected to  $v[i]$   
            //  $0 \leq j < k$ , if there are  $k$  vertices adjacent to  $v[i]$   
            {  
                if (the geometry of  $a[j]$  is interpolated or is not lost) {  
                    geometry_sum +=  $a[j].geometry$ ;  
                    i++;}  
            }  
         $v[i].geometry = geometry\_sum / i$ ;  
        label  $v[i]$  as interpolated;  
    }  
}
```

REFERENCES

- [1] G. Alregib and Y. Altunbasak, "3TP: An application-layer protocol for streaming 3D models," *IEEE Transactions on Multimedia*, pp. 1149-1156, December 2006.
- [2] G. Alregib, Y. Altunbasak and J. Rossignac, "Error-Resilient Transmission of 3D Models," *ACM Trans. on Graphics*, April 2005. (Early version in ICASSP 02.)
- [3] P. Alliez and M. Desbrun, "Progressive Encoding for Lossless Transmission of Triangle Meshes," *SIGGRAPH*, 2001, pages 198-205.

- [4] P. Alliez and M. Desbrun. "Valence-Driven Connectivity Encoding of 3D Meshes," *EUROGRAPHICS*, 2001, pages 480-489.
- [5] A. Bakre and B. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," *Int. Conf. Distributed Computing Systems*, 1995, pages.136–143.
- [6] I. Cheng and P. Boulanger, "Just Noticeable Difference for 3D Perception," *Eurographics*, Dubin, August 2005.
- [7] I. Cheng and P. Boulanger, "Feature Extraction on 3D TexMesh using Scale-space Analysis and Perceptual Evaluation," *IEEE Trans. on CSVT*, Special Issue on Scale-space Feature Extraction, Oct, 2005.
- [8] I. Cheng and A. Basu, "Optimal aspect ratio for 3D TV," IEEE 3D TV Conference, 4 proceedings pages, KOS, Greece, May 2007.
- [9] I. Cheng and A. Basu, "Perceptually optimized 3D transmission over wireless networks," *IEEE Trans. on Multimedia*, 386-396, February 2007.
- [10] I. Cheng, L. Ying and A. Basu, "A Perceptually Driven Model for Transmission of Arbitrary 3D Models over Unreliable Networks," *3rd Intl Symposium on 3DPVT*, June 2006, North Carolina, USA.
- [11] Z. Chen, B. Bodenheimer and J. Barnes, "Robust Transmission of 3D Geometry over Wireless Networks," In *Web3D*, 2003, pages 161-172.
- [12] R. Caceres and L. Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," *IEEE J. Select. Areas Comm.*, vol. 13, pp. 850–857, June 1995.
- [13] E. Catmull and J. Clark, "Recursively Generated B-spline Surfaces on Arbitrary Topological Surfaces," *Computer-Aided Design*, 10(6):350-355, November 1978.
- [14] P. Cignoni, C. Rocchini and R. Scopigno, "Metro: measuring error on simplified surfaces," *Computer Graphics Forum*, Blackwell Publishers, vol. 17(2), June 1998, pp 167-174.
- [15] A. Conti, D. Dardari, G. Pasolini and O. Andrisano, "Bluetooth and IEEE 802.11b coexistence: analytical performance evaluation in fading channels," *IEEE Journal on Selected Areas in Communications*, pp. 259- 269, Vol.21-2, Feb 2003.
- [16] M. Deering, "Geometric Compression," *SIGGRAPH*, 1995.

- [17] D. Farin, R. Peerlings and P. H. N. de With, "Depth-image representation employing meshes for intermediate-view rendering and coding," IEEE 3D TV Conference, 4 proceedings pages, KOS, Greece, May 2007.
- [18] MathWorks, <http://www.mathworks.com/access/helpdesk/help/techdoc/ref/griddata.html>.
- [19] A. Hać and A. Armstrong, "Resource allocation scheme for QoS provisioning in microcellular networks carrying multimedia traffic," *Int. Journal of Network Management*, vol. 11, 277 – 307, 2001.
- [20] A. Hać and Z. Chen, "A hybrid channel allocation method for wireless communication networks," *Int. Journal of Network Management*, vol. 10, 59-74, 2000.
- [21] A. Hać and B. Liu, "Database and Location Management Schemes for Mobile Communications," *IEEE/ACM Trans. on Networking*, Vol. 6, No. 6, 851-865, Dec. 1998.
- [22] P. Heckbert and M. Garland, *Survey of Polygonal Surface Simplification Algorithms*, TR, Carnegie Mellon University, 1997.
- [23] M. Hassan, M. M. Krunz and I. Matta, "Markov-Based Channel Characterization for Tractable Performance Analysis in Wireless Packet Networks," *IEEE Trans. on Wireless Communications*, May 2004.
- [24] H. Hoppe, "Progressive Meshes," *SIGGRAPH*, 1996.
- [25] P. Jaromersky, X. Wu, and Y. Chiang, "Multiple-Description Geometry Compression for Networked Interactive 3D Graphics," In *International Conference on Image and Graphics*, 2005.
- [26] M. Kazhdan, "Reconstruction of Solid Models from Oriented Point Sets," *Eurographics Symposium on Geometry Processing*, 2005.
- [27] L. Kobbelt, "Sqrt(3)-Subdivision," *SIGGRAPH*, 2000.
- [28] K. Lee and S. Chanson, "Packet Loss Probability for Real-time Wireless Communications," *IEEE Trans. on Vehicular Technology*, Nov. 2002.
- [29] T. Lewiner, H. Lopes, J. Rossignac and A. Vieira, "Efficient Edgebreaker for Surfaces of Arbitrary Topology," *SIGGRAPH*, 2004.
- [30] D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner, *Level of Detail for 3D Graphics*, Morgan Kaufmann, 2002.
- [31] J. O. Limb, "Distortion Criteria of the Human Viewer," *IEEE Transactions on SMC*, 778-793, 1979.

- [32] J. L. Mannos and D. J. Sakrison, "The Effects of a Visual Fidelity Criterion on Encoding of Images," *IEEE Trans. on Information Theory*, 1974.
- [33] F. Mazzenga, D. Cassioli, P. Loreti and F. Vatalaro, "Evaluation of Packet Loss Probability in Bluetooth Networks," *IEEE International Conference on Communications (ICC'02)*, New York, NY, USA, 2002.
- [34] F. Mazzenga, D. Cassioli, A. Detti, I. Habib, P. Loreti and F. Vatalaro, "Performance Evaluation in Bluetooth Dense Piconet Areas", *IEEE Transactions on Wireless Communications*, Vol. 3, No. 6, November 2004.
- [35] ISO/IEC 14496-2:2000, Amendment 1. Coding of Audio-Visual Objects-Part 2: Visual, Version 2, 2000.
- [36] Y. Pan, I. Cheng and A. Basu, "Quality Metric for Approximating Subjective Evaluation of 3D Objects," *IEEE Trans. on Multimedia*, 269-279, April 2005. (Short version in ICIP 2003.)
- [37] F. Pereira and T. Ebrahimi, *The MPEG-4 Book*, Prentice-Hall, Upper Saddle River, New Jersey, 2002.
- [38] J. Peng, C.-S. Kim, and C.-C. Kuo, "Technologies for 3D Mesh Compression: A Survey," *Journal of Visual Communication and Image Representation*, 16 (2005) 688-733, 2005.
- [39] R.J. Punnoose, R.S. Tseng, D.D. Stancil, "Experimental Results for Interference between Bluetooth and IEEE 802.11 b DSSS Systems," *IEEE Vehicular Technology Conference*, 2001.
- [40] D.T. Sandwell, "Biharmonic Spline Interpolation of GEOS-3 and SEASAT Altimeter Data," *Geophysical Research Letters*, 14, 2, 139-142, 1987.
- [41] O. Sorkine, D. Cohen-Or, and S. Toledo, "High-pass quantization for mesh encoding," Eurographics Symposium on Geometry processing, 2003.
- [42] G. Regib and Y. Altunbasak, "An Unequal Error Protection Method for Packet Loss Resilient 3D Mesh Transmission," *INFOCOM*, 2002.
- [43] J. Rossignac, "Edgebreaker: Connectivity compression for triangle meshes," *IEEE Trans. on Visualization and Computer Graphics*, 1999, 47-61.
- [44] J. Rossignac, "Surface Simplification and 3D Geometry Compression", In *Handbook of Discrete and Computational Geometry* (second edition), CRC Press, 2003.
- [45] D. Tian and G. AlRegib, "FQM: A fast quality measure for efficient transmission of textured 3D models," *ACM Multimedia* 2004.

- [46] G. Taubin, "3D Geometry Compression and Progressive Transmission," In *Proc. EUROGRAPHICS*, 1999.
- [47] C. Touma and C. Gotsman, "Triangle Mesh Compression," In *Graphics Interface*, 1998, pages 26-34.
- [48] G. Taubin, A. Gujãeziec, W. Horn and F. Lazarus, "Progressive Forest Split Compression," In *Proc. SIGGRAPH*, 1998.
- [49] G. Taubin and J. Rossignac, "Geometric Compression Through Topological Surgery," In *Proc. SIGGRAPH*, 1998.
- [50] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.
- [51] The Virtual Reality Modeling Language (VRML). ISO/IEC 14772-1, 1997.
- [52] G. Wittenburg and J. Schiller, "A Quantitative Evaluation of the Simulation Accuracy of Wireless Sensor Networks, " In *Proceedings of 6. Fachgesprach "Drahtlose Sensornetze" der GI/ITG-Fachgruppe "Kommunikation und Verteilte Systeme"*, Aachen, Germany, July 2007.
- [53] D. Wu and R. Negi, "A Wireless Channel Model for Support of Quality of Service," *IEEE Global Communication Conference (GLOBECOM)*, San Antonio, Texas, USA, Nov. 25-29, 2001.
- [54] D. Wu and R. Negi, "Effective Capacity: A Wireless Channel Model for Support of QoS," *IEEE Trans. on Wireless Communications*, 2002.
- [55] X. Xiang, M. Held and J. Mitchell, "Fast & Effective Stripification of Polygonal Surface Models", In *Proc. ACM Symposium on Interactive 3D Graphics (I3DG'99)*, Atlanta, USA.
- [56] S. Yang *et al.* "Optimized mesh and texture multiplexing for progressive textured model transmission," *ACM Multimedia*, 2004.
- [57] Z. Yan, S. Kumar, and C-C. Kuo, "Error-Resilient Coding of 3-D Graphic Models via Adaptive Mesh Segmentation," *IEEE Transactions on CSVT*, July 2001.
- [58] L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks," *IEEE Networks special issue on network security*, November/December, 1999.