9-2-2008

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

Irene Wan Cheng
University of Pennsylvania

Lihang Ying
University of Alberta

Kostas Daniilidis
University of Pennsylvania, kostas@cis.upenn.edu

Anup Basu
University of Alberta

Irene Cheng, Lihang Ying, Kostas Daniilidis, and Anup Basu, "Robust and Scalable Transmission of Arbitrary 3D Models over Wireless Networks," EURASIP Journal on Image and Video Processing, vol. 2008, Article ID 890482, 14 pages, 2008. doi:10.1155/2008/890482

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

Abstract

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 versus 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 retransmitted. Only the valence (connectivity) packets need to be retransmitted; however, storage of valence information requires only 10% space compared to vertices and even less compared to photorealistic texture. Thus, less than 5% of the packets may need to be retransmitted 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.

Comments

Irene Cheng, Lihang Ying, Kostas Daniilidis, and Anup Basu, "Robust and Scalable Transmission of Arbitrary 3D Models over Wireless Networks," EURASIP Journal on Image and Video Processing, vol. 2008, Article ID 890482, 14 pages, 2008. doi:10.1155/2008/890482

Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 890482, 14 pages
doi:10.1155/2008/890482

Research Article

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

Irene Cheng,[1, 2] Lihang Ying,[2] Kostas Daniilidis,[1] and Anup Basu[2]

[1] *Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389, USA*
[2] *Department of Computing Science, University of Alberta, Edmonton, AB, Canada T6G 2E8*

Correspondence should be addressed to Irene Cheng, chenglin@seas.upenn.edu

Received 26 February 2008; Revised 15 July 2008; Accepted 2 September 2008

Recommended by Peter Eisert

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 versus 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 retransmitted. Only the valence (connectivity) packets need to be retransmitted; however, storage of valence information requires only 10% space compared to vertices and even less compared to photorealistic texture. Thus, less than 5% of the packets may need to be retransmitted 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.

1. INTRODUCTION

The benefit of designing packets optimizing texture-mesh tradeoff was discussed in our earlier work [1]. 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 [2, 3]. 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 3000 left, 1000 middle, and 500 right). There exists substantial literature on multimedia transmission over wireless networks, such as [4, 5]. 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 [6, 7]. In [8, 9], 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 [10], and some retransmission may be necessary. Also, some of the approaches proposed earlier assume bit error correction rather than lost packets. Packet loss probability models have been proposed by some researchers, for example, [11]; 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 [1] 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.

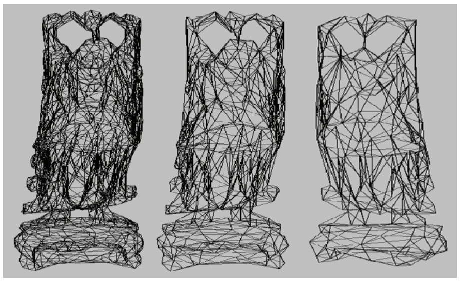


Figure 1: Buddha model at various mesh resolution levels.

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 [12] (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 [13]. 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 [14] 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, 16] 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 et al. [17, 18] 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 $20*20\,\mathrm{m}^2$, the packet loss probability could be up to 60%.

The authors in [8] consider packet loss up to 40%. In [9], partial data is transmitted by UDP, and the work considers the situation of receiving 300 000 faces out of 1.08 M faces, which is equivalent to more than 70% packet loss. In the context of multidescription transmission [19], only 1 out of 4 descriptions is considered to be transmitted due to limited bandwidth. In multicast or broadcast situation, no acknowledgement or 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 [20–24]; 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 [25–28], described in detail in [1, 29], 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.

2. 3D MESH CODING FOR TRANSMISSION

A 3D mesh is represented by geometry and connectivity [30]. An uncompressed representation, such as the VRML ASCII format [13], 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.

Among the many 3D mesh compression schemes proposed since the early 1990s [31], the valence-driven approach [32] is considered to be a state-of-the-art technique [33, 34] 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 [31]. A number of 3D mesh compression algorithms have been accepted as international standards. For example, topological surgery [35] and progressive forest split [36] have been adopted by VRML version 2 [37] and MPEG-4 version 2, defined as 3D mesh coding (3DMC) [38].

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 (see 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.

There are many other innovative approaches for mesh and connectivity coding and compression, including topological surgery [35], progressive forest split [36], MPEG-4 3D mesh coding (3DMC) [39], and so on. A detailed review of these papers can be found in [29], and is thus not included here.

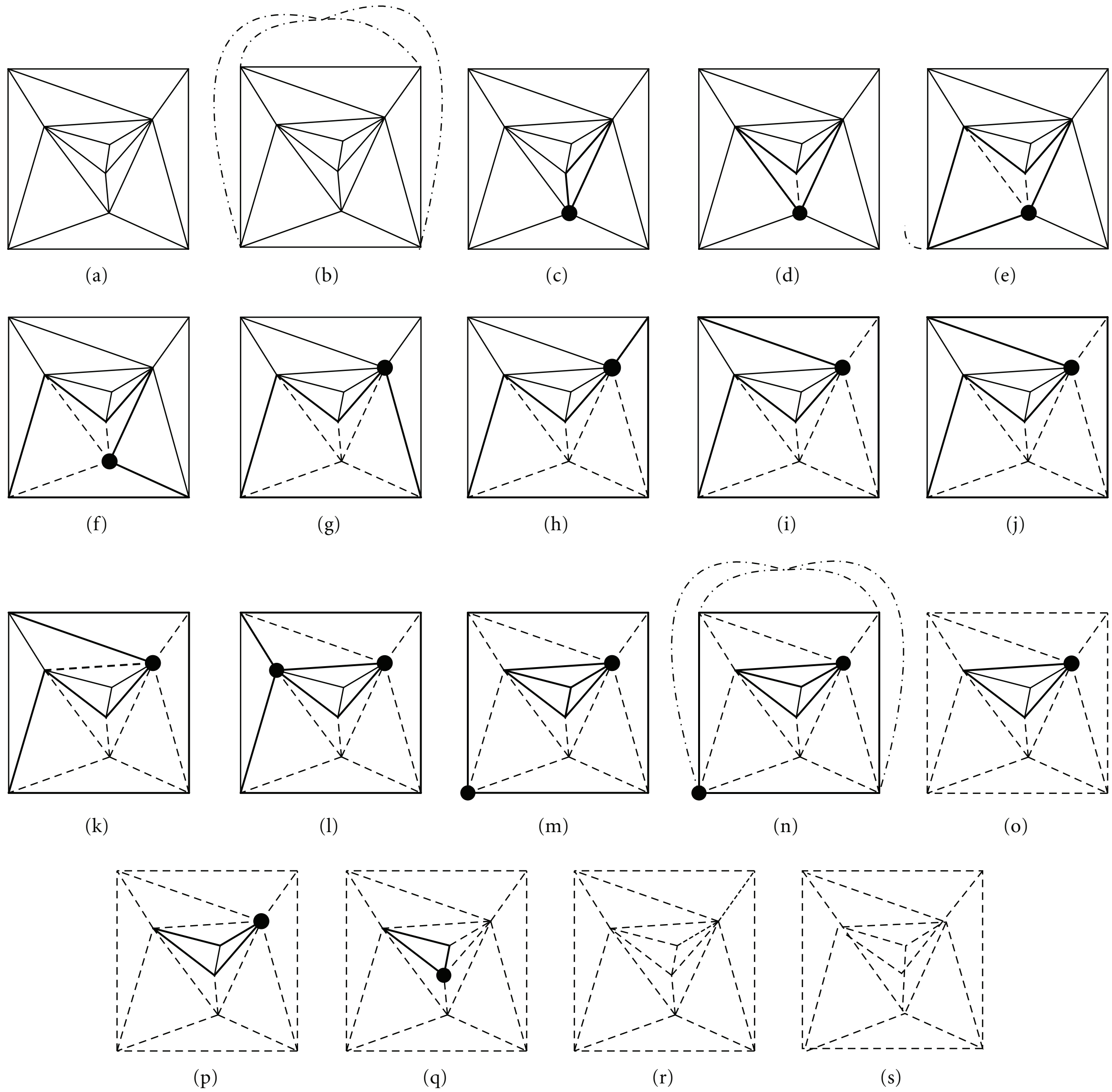


Figure 2: An example of a run of the valence-driven connectivity encoding algorithm. The active lists are indicated by thick lines and edges already visited (encoded) by dashed lines.

Current 3D mesh coding techniques mainly focus on coding efficiency, that is, 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 [40, 41]. With one error character in the connectivity stream, the decoded mesh can change significantly and can be impossible to reconstruct.

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. Reference [42] 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. Reference [19] 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 [8, 43].

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

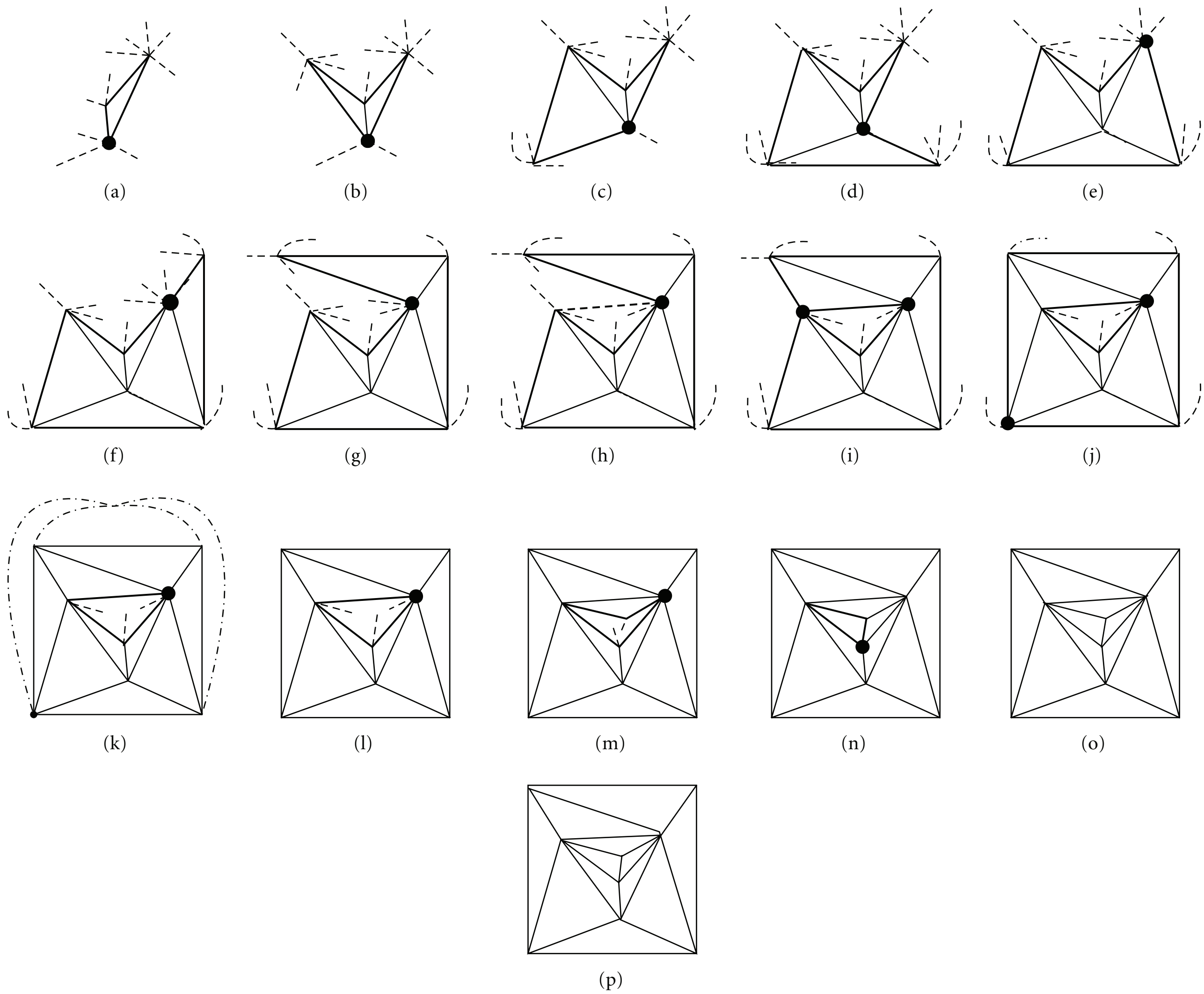


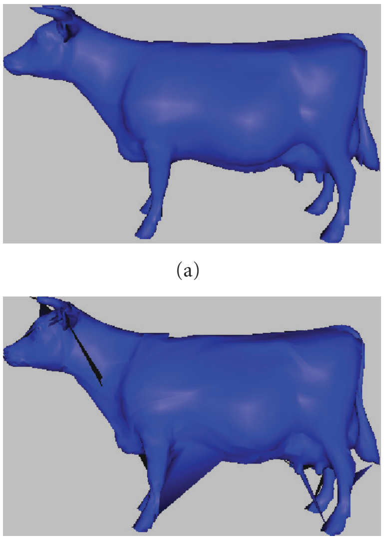
Figure 3: An example of connectivity decoding (or reconstructing) from the stream of vertex valences in the valence-driven algorithm.

transmission of regular meshes can be extended to work with irregular meshes.

3. 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 semiregular 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 predetermined 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 [44] of the base layers to safeguard the successful transmission of important features of the object [8, 45]. In progression-based algorithms [30, 33, 36, 46–48], 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 tradeoff between compression rate and robustness to packet loss. For example, although the Edgebreaker 3D mesh coding method discussed in Section 2 has high compression ratio, the cow object (see Figure 4) shows significant distortion even when one character in the connectivity chain is lost. In our strategy, we focus on the following criteria.



(a)

(b)

Figure 4: An example of error sensitivity of the Edgebreaker 3D mesh coding method. (a) Original 3D mesh, (b) decoded 3D mesh with one error character in the decoded connectivity stream.

Figure 5: Applying stripification to a cow mesh [50]. Different colors represent different triangle strips. (http://www.cosy.sbg.ac.at/~held/projects/strips/strips.html)

(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 [49, 50] 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.

(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, that is, if packet(s) containing valence information are lost they are retransmitted.

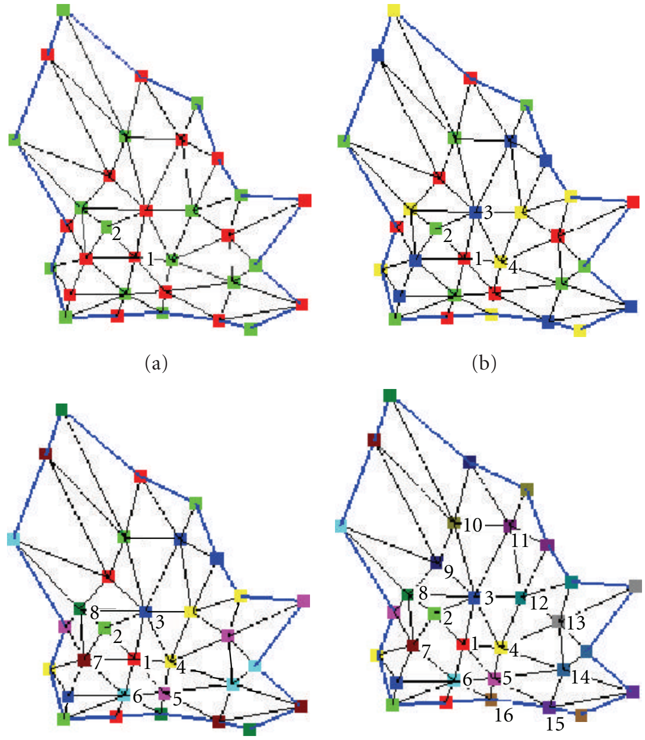


Figure 6: (a) 2 packets, (b) 4 packets, (c) 8 packets, (d) 16 packets.

(3) Texture-mesh tradeoff based on perceptual optimization

This topic will be an extension of our previous work [1, 29] and will be considered in future work.

3.1. 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.

3.2. Interpolation of lost geometry

After all packets are received, first, the mesh is partially reconstructed based on the geometry packets received and

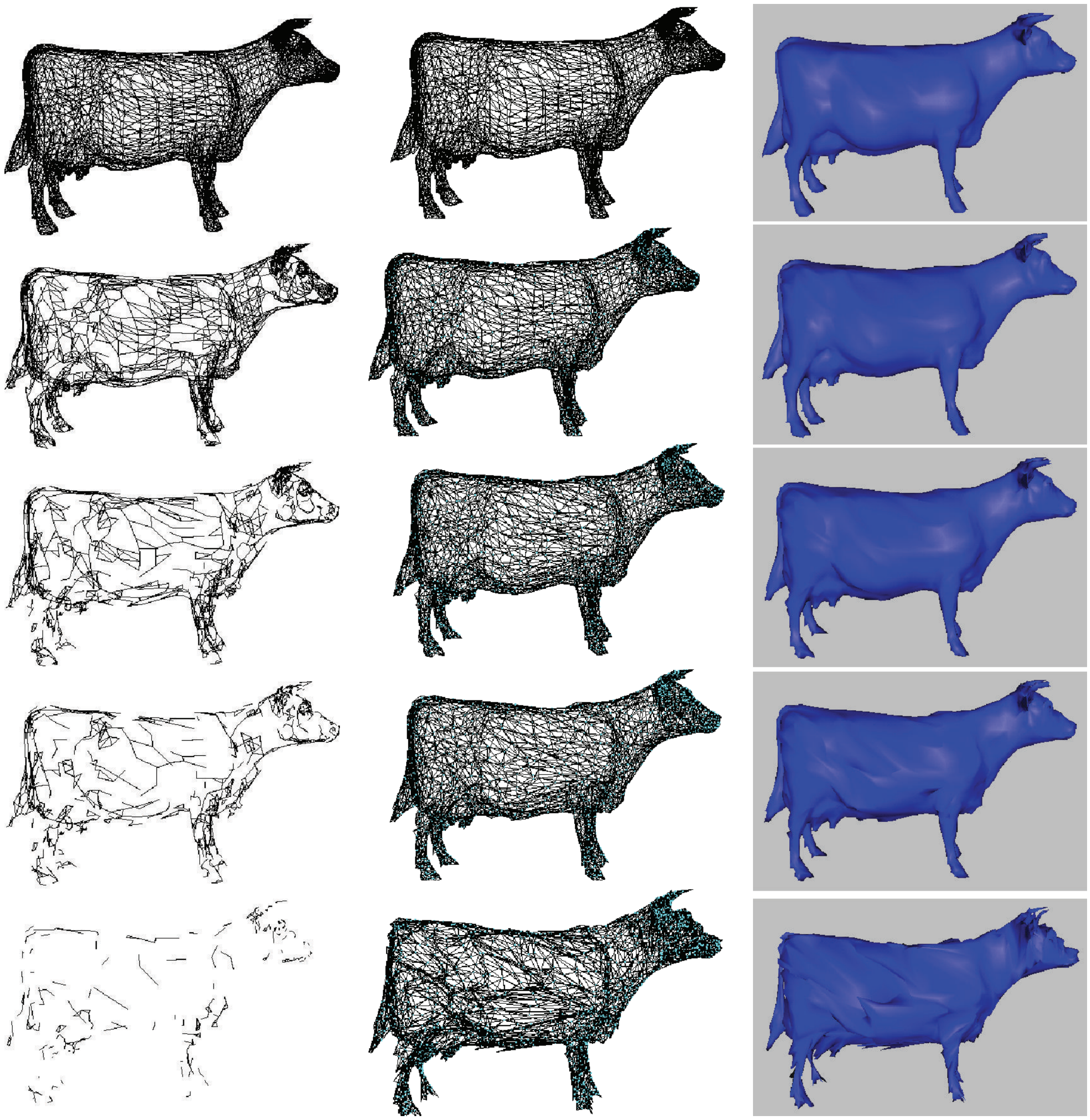


Figure 7: From top to bottom, (a) 0%, 30%, 50%, 60%, and 80% randomly selected packet loss was applied to a cow mesh; (b) interpolated meshes, (c) the corresponding mesh mapped with color.

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 not lost or interpolated previously, are used to interpolate the geometry of L. Several interpolation strategies, linear, cubic, and Laplacian were considered. Brief pseudocode of an interpolation method is given in the appendix.

4. 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 [31], 13.3 bits/vertex to encode geometry [51], and 650 vertices and 50 Kbytes or higher for the compressed photorealistic textures in Figure 9, the cost of retransmission of the connectivity information for this real example is less than 1%. Thus, to avoid the delays in requesting retransmission of packets, it may be wiser to send duplicate packets containing the connectivity information so that real-time visualization of photorealistic texture mapped 3D objects at high packet loss can be facilitated.

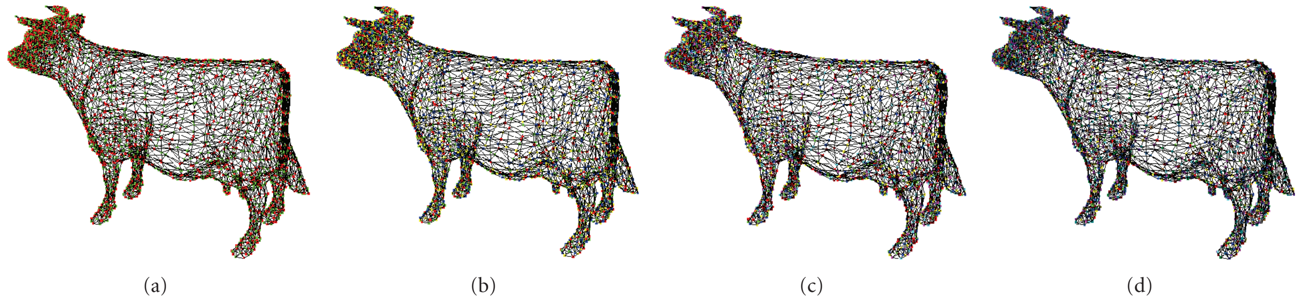


Figure 8: Cow vertices encoded in (a) 2 packets, (b) 4 packets, (c) 8 packets, (d) 16 packets.

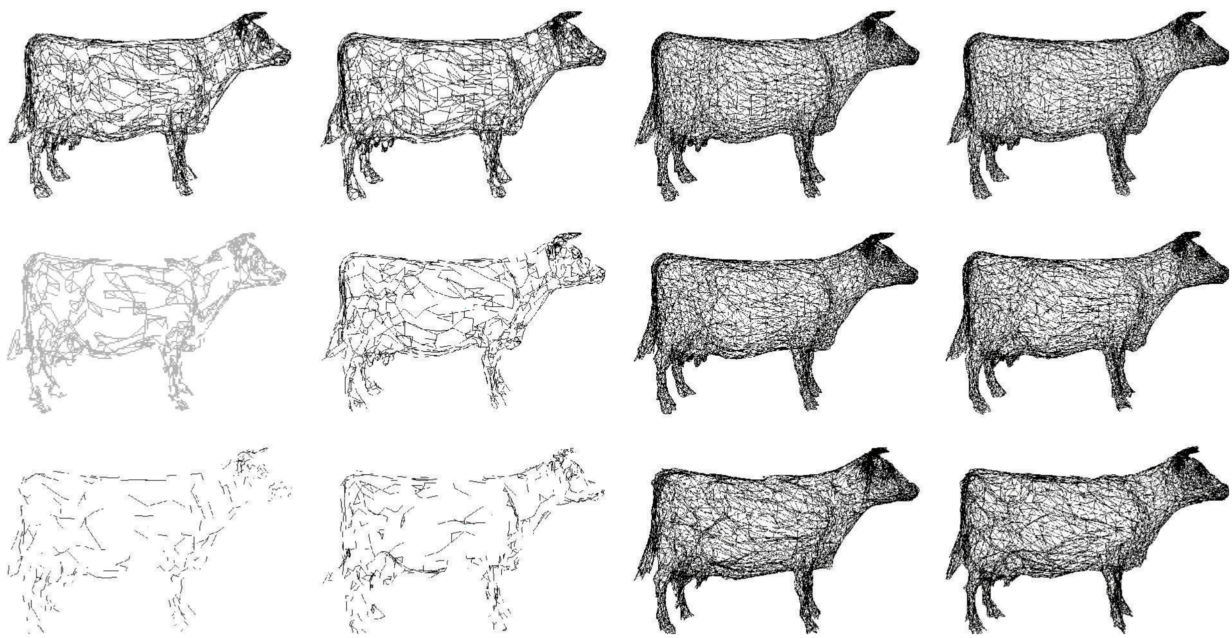


Figure 9: From top to bottom: (*Column 1, before interpolation*): 4 out of 16 packets lost; 8 out of 16 packets lost; 12 out of 16 packets lost; (*Column 2, before interpolation*): 1 out of 4 packets lost; 2 out of 4 packets lost; and 3 out of 4 packets lost; (*Column 3, after interpolation*): 4 out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

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 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.

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 (see Figure 11) compared to results obtained after much higher loss by our method (see Figure 9).

Some videos of our implementation results can be seen at http://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.

5. COMPARISON OF DIFFERENT INTERPOLATION METHODS

We applied the triangle-based linear, triangle-based cubic spline, and "v4" [52] interpolation methods [53] 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 [54] 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

Table 1: 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.)

(a) Number of lost packets (out of 16) = 4

	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)	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

(b) Number of lost lackets (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

(c) 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

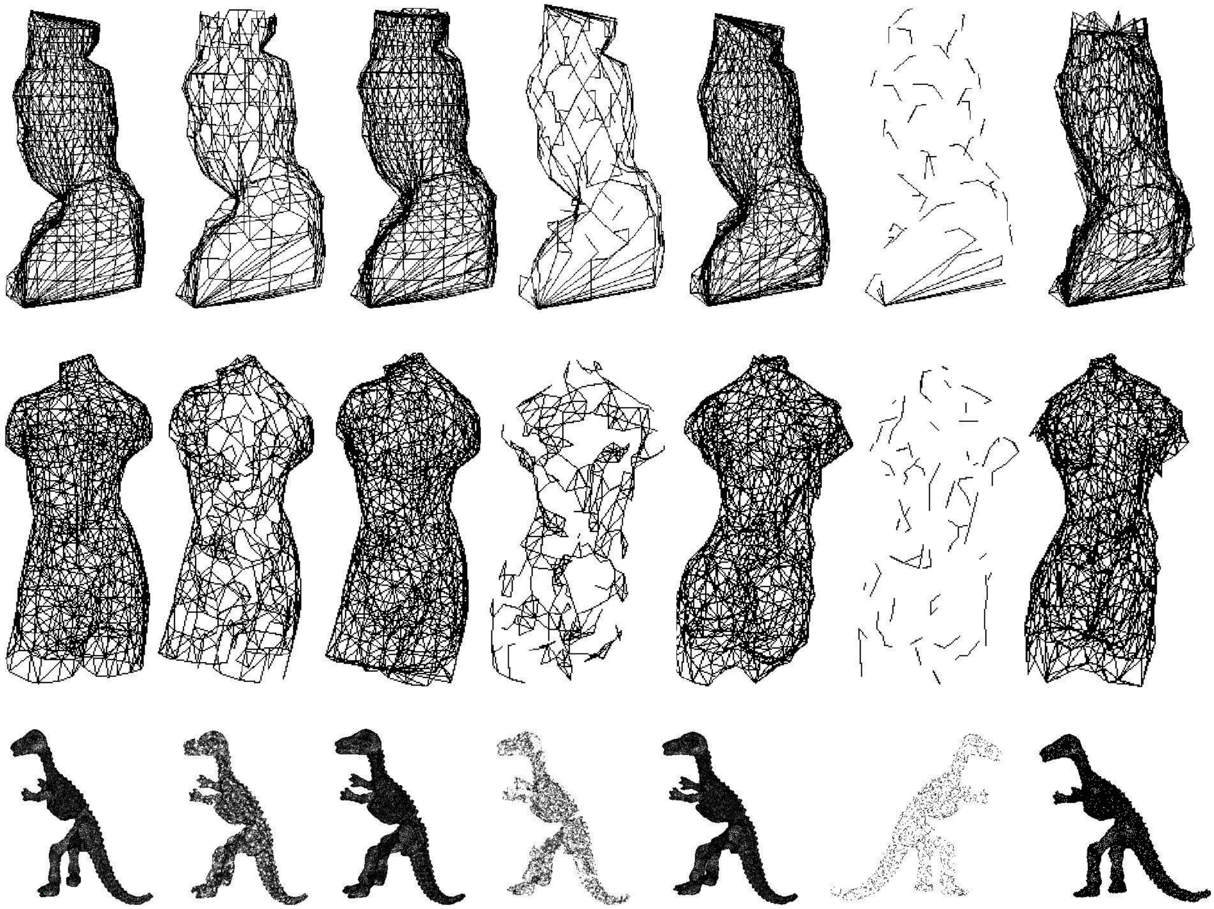
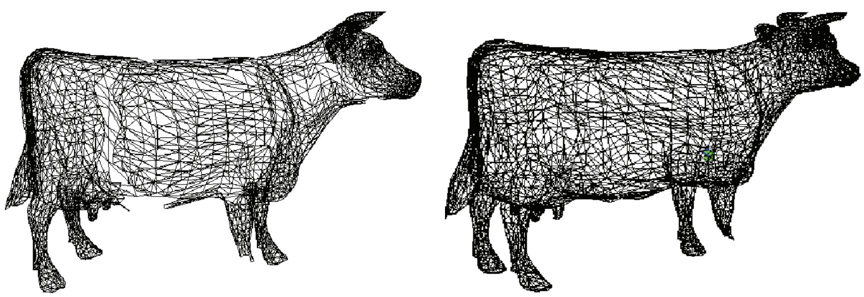


FIGURE 10: Different models: queen (1st row, 650 vertices); body (2nd row, 711 vertices); dinosaur (3rd row, 14070 vertices). 1st column: original model; 2nd column: 4 loss out of 16 packets (before interpolation); 3rd column: 4 loss out of 16 packets (after interpolation); 4th column: 8 loss out of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



(a) (b)

FIGURE 11: Effect of packet loss when nearby vertices are not distributed into different packets (1 out of 16 packets loss). (a) Before interpolation, (b) After interpolation.

a number of samples that is approximately 10 times the number of faces.

In Table 1, 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

Table 2: 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.

	Reconstruction error			
	No. of lost packets (out of 16) = 1		No. of lost packets (out of 16) = 2	
Model (vertex number)	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(*)

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.

5.1. *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 [55]—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 [56] and sqrt(3)-subdivision method [57]. 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 [56], the coordinates of added vertices are interpolated following the cubic spline algorithm. Sqrt(3)-subdivision method [57] differs from other subdivision methods by increasing the number of triangles in every step by a factor of 3 instead of 4.

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 2 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 (see Table 3).

Table 3: Comparison among different subdivision methods and subdivision steps. The test model is cow.

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

Sorkine et al. [58] 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.

6. EFFECT OF PACKETIZATION ON MESH COMPRESSION

In order to support packet loss scenarios, in our scheme, each packet is compressed independently following the

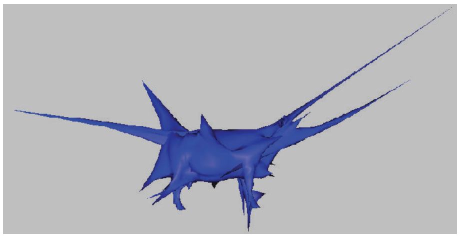


Figure 12: The reconstructed cow model (2904 vertices) after losing 2% of the Laplacian values.

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.

7. 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 is 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

OUTLINE OF AN INTERPOLATION ALGORITHM

A linear interpolation algorithm can be described as in Algorithm 1.

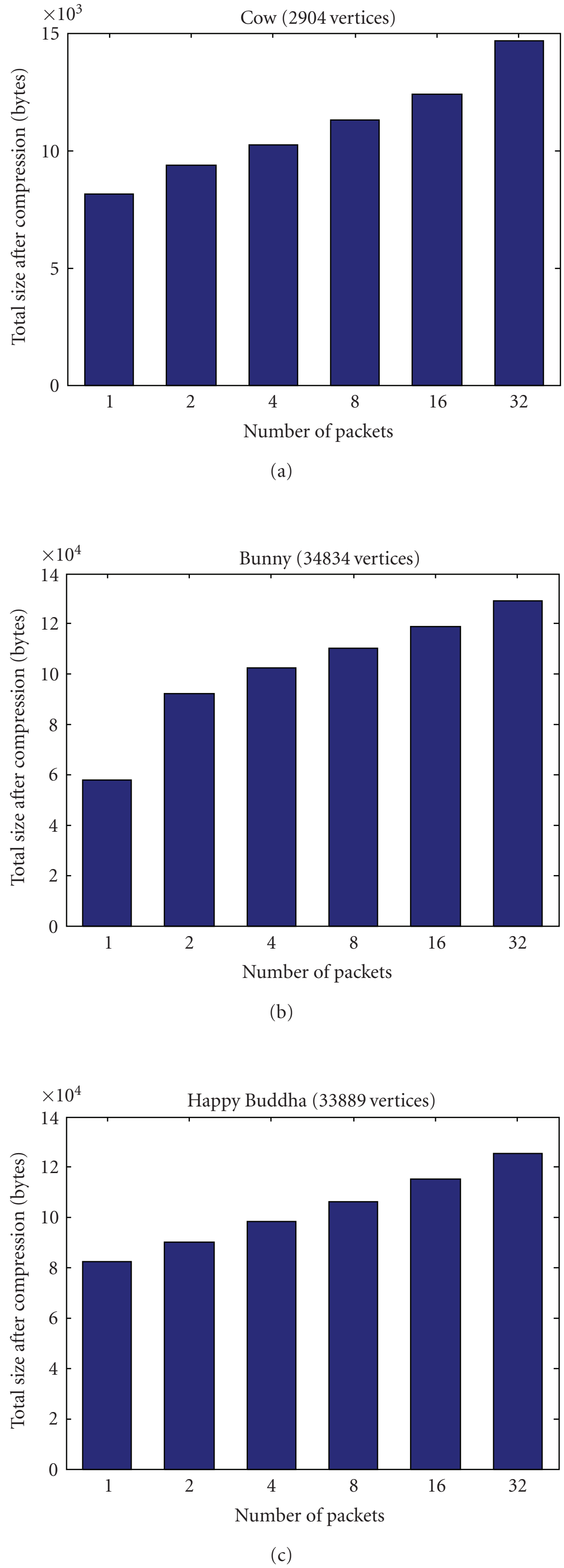


Figure 13: Total data size after compression considering different numbers of packets: (a) cow (2904 vertices); (b) bunny (34834 vertices); (c) happy Buddha (33889 vertices).

```
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 <= 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;
        }
    }
```

Algorithm 1

ACKNOWLEDGMENTS

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

REFERENCES

[1] I. Cheng and A. Basu, "Perceptually optimized 3D transmission over wireless networks," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 386–396, 2007.

[2] P. Heckbert and M. Garland, "Survey of polygonal surface simplification algorithms," Tech. Rep., Carnegie Mellon University, Pittsburgh, Pa, USA, 1997.

[3] D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner, *Level of Detail for 3D Graphics*, Morgan Kaufmann, San Francisco, Calif, USA, 2002.

[4] D. Wu and R. Negi, "A wireless channel model for support of quality of service," in *Proceedings of the IEEE Global Communication Conference (GLOBECOM '01)*, vol. 1, pp. 695–699, San Antonio, Tex, USA, November 2001.

[5] D. Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 630–643, 2003.

[6] I. Cheng and A. Basu, "Optimal aspect ratio for 3D TV," in *Proceedings of the IEEE 3DTV Conference*, pp. 1–4, Kos Island, Greece, May 2007.

[7] D. Farin, R. Peerlings, and P. H. N. de With, "Depth-image representation employing meshes for intermediate-view rendering and coding," in *Proceedings of the IEEE 3DTV Conference*, pp. 1–4, Kos Island, Greece, May 2007.

[8] G. AlRegib, Y. Altunbasak, and J. Rossignac, "Error-resilient transmission of 3D models," *ACM Transactions on Graphics*, vol. 24, no. 2, pp. 182–208, 2005.

[9] Z. Chen, B. Bodenheimer, and J. F. Barnes, "Robust transmission of 3D geometry over lossy networks," in *Proceedings of the 8th International Conference on 3D Web Technology (Web3D '03)*, pp. 161–172, Saint Malo, France, March 2003.

[10] G. AlRegib and Y. Altunbasak, "3TP: an application-layer protocol for streaming 3D models," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1149–1156, 2006.

[11] K. K. Lee and S. T. Chanson, "Packet loss probability for real-time wireless communications," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 6, pp. 1569–1575, 2002.

[12] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[13] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, Cambridge, UK, 2005.

[14] G. Wittenburg and J. Schiller, "A quantitative evaluation of the simulation accuracy of wireless sensor networks," in *Proceedings of the 6th Fachgespräch "Drahtlose Sensornetze" der GI/ITG-Fachgruppe "Kommunikation und Verteilte Systeme"*, pp. 23–26, Aachen, Germany, July 2007.

[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*, vol. 21, no. 2, pp. 259–269, 2003.

[16] R. J. Punnoose, R. S. Tseng, and D. D. Stancil, "Experimental results for interference between Bluetooth and IEEE 802.11b DSSS systems," in *Proceedings of the 54th IEEE VTS Vehicular Technology Conference (VTC '01)*, vol. 1, pp. 67–71, Atlantic City, NJ, USA, October 2001.

[17] F. Mazzenga, D. Cassioli, P. Loreti, and F. Vatalaro, "Evaluation of packet loss probability in Bluetooth networks," in *Proceedings of the IEEE International Conference on Communications (ICC '02)*, vol. 1, pp. 313–317, New York, NY, USA, April-May 2002.

[18] 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, pp. 2362–2373, 2004.

[19] P. Jaromersky, X. Wu, Y.-J. Chiang, and N. Memon, "Multiple-description geometry compression for networked interactive 3D graphics," in *Proceedings of the 3rd International Conference on Image and Graphics (ICIG '04)*, pp. 468–471, Hong Kong, December 2004.

[20] A. Bakre and B. R. Badrinath, "I-TCP: indirect TCP for mobile hosts," in *Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS '95)*, pp. 136–143, Vancouver, Canada, May-June 1995.

[21] A. Hać and A. Armstrong, "Resource allocation scheme for QoS provisioning in microcellular networks carrying multimedia traffic," *International Journal of Network Management*, vol. 11, no. 5, pp. 277–307, 2001.

[22] A. Hać and Z. Chen, "A hybrid channel allocation method for wireless communication networks," *International Journal of Network Management*, vol. 10, no. 2, pp. 59–74, 2000.

[23] A. Hać and B. Liu, "Database and location management schemes for mobile communications," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 851–865, 1998.

[24] M. Hassan, M. M. Krunz, and I. Matta, "Markov-based channel characterization for tractable performance analysis in wireless packet networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 3, pp. 821–831, 2004.

[25] I. Cheng and P. Boulanger, "A 3D perceptual metric using just-noticeable-difference," in *Proceedings of the Annual Conference of the European Association for Computer Graphics (Eurographics '05)*, pp. 97–100, Dublin, Ireland, August-September 2005.

[26] I. Cheng and P. Boulanger, "Feature extraction on 3D TexMesh using scale-space analysis and perceptual evaluation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1234–1244, 2005.

[27] J. L. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion of the encoding of images," *IEEE Transactions on Information Theory*, vol. 20, no. 4, pp. 525–536, 1974.

[28] Y. Pan, I. Cheng, and A. Basu, "Quality metric for approximating subjective evaluation of 3D objects," *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 269–279, 2005.

[29] I. Cheng, L. Ying, and A. Basu, "A perceptually driven model for transmission of arbitrary 3D models over unreliable networks," in *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT '06)*, pp. 421–428, Chapel Hill, NC, USA, June 2006.

[30] D. Tian and G. AlRegib, "FQM: a fast quality measure for efficient transmission of textured 3D models," in *Proceedings of the 12th Annual ACM International Conference on Multimedia (MM '04)*, pp. 684–691, New York, NY, USA, October 2004.

[31] J. Peng, C.-S. Kim, and C.-C. Jay Kuo, "Technologies for 3D mesh compression: a survey," *Journal of Visual Communication and Image Representation*, vol. 16, no. 6, pp. 688–733, 2005.

[32] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proceedings of the Graphics Interface (GI '98)*, pp. 26–34, Vancouver, Canada, June 1998.

[33] P. Alliez and M. Desbrun, "Progressive compression for lossless transmission of triangle meshes," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 195–202, Los Angeles, Calif, USA, August 2001.

[34] P. Alliez and M. Desbrun, "Valence-driven connectivity encoding of 3D meshes," in *Proceedings of the Eurographics Workshop (Eurographics '01)*, pp. 480–489, Manchester, UK, September 2001.

[35] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, Orlando, Fla, USA, July 1998.

[36] G. Taubin, A. Guéziec, W. Horn, and F. Lazarus, "Progressive forest split compression," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, pp. 123–132, Orlando, Fla, USA, July 1998.

[37] ISO/IEC 14772-1, "The Virtual Reality Modeling Language (VRML)," 1997.

[38] ISO/IEC 14496-2:2000, Amendment 1, "Coding of Audio-Visual Objects—Part 2: Visual version 2," 2000.

[39] F. Pereira and T. Ebrahimi, *The MPEG-4 Book*, Prentice-Hall, Upper Saddle River, NJ, USA, 2002.

[40] T. Lewiner, H. Lopes, J. Rossignac, and A. W. Vieira, "Efficient Edgebreaker for surfaces of arbitrary topology," in *Proceedings of the 17th Brazilian Symposium on Computer Graphics and Image Processing (SIGGRAPH '04)*, pp. 218–225, Curitiba, Brazil, October 2004.

[41] J. Rossignac, "Edgebreaker: connectivity compression for triangle meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 47–61, 1999.

[42] Z. Yan, S. Kumar, and C.-C. Kuo, "Error-resilient coding of 3D graphic models via adaptive mesh segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 7, pp. 860–873, 2001.

[43] G. AlRegib and Y. Altunbasak, "An unequal error protection method for packet loss resilient 3D mesh transmission," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 2, pp. 743–752, New York, NY, USA, June 2002.

[44] R. Caceres and L. Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 5, pp. 850–857, 1995.

[45] J. O. Limb, "Distortion criteria of the human viewer," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 12, pp. 778–793, 1979.

[46] H. Hoppe, "Progressive meshes," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pp. 99–108, New York, NY, USA, August 1996.

[47] G. Taubin, "3D geometry compression and progressive transmission," in *EUROGRAPHICS—State of the Art Report*, pp. 81–96, Milano, Italy, September 1999.

[48] S. Yang, C.-H. Lee, and C.-C. Jay Kuo, "Optimized mesh and texture multiplexing for progressive textured model transmission," in *Proceedings of the 12th Annual ACM International Conference on Multimedia (MM '04)*, pp. 676–683, New York, NY, USA, October 2004.

[49] M. Deering, "Geometry compression," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, pp. 13–20, New York, NY, USA, September 1995.

[50] X. Xiang, M. Held, and J. S. B. Mitchell, "Fast and effective stripification of polygonal surface models," in *Proceedings of the Symposium on Interactive 3D Graphics (I3DG '99)*, pp. 71–78, Atlanta, Ga, USA, April-May 1999.

[51] J. Rossignac, "Surface simplification and 3D geometry compression," in *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, Fla, USA, 2nd edition, 2003.

[52] D. T. Sandwell, "Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data," *Geophysical Research Letters*, vol. 14, no. 2, pp. 139–142, 1987.

[53] MathWorks, http://www.mathworks.com/access/helpdesk/help/techdoc/ref/griddata.html.

[54] P. Cignoni, C. Rocchini, and R. Scopigno, "*Metro*: measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998.

[55] M. Kazhdan, "Reconstruction of solid models from oriented point sets," in *Proceedings of the 3rd Eurographics Symposium on Geometry Processing (SGP '05)*, pp. 73–82, Vienna, Austria, July 2005.

[56] E. Catmull and J. Clark, “Recursively generated B-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, no. 6, pp. 350–355, 1978.

[57] L. Kobbelt, “$\sqrt{3}$-subdivision,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’00)*, pp. 103–112, New Orleans, La, USA, July 2000.

[58] O. Sorkine, D. Cohen-Or, and S. Toledo, “High-pass quantization for mesh encoding,” in *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP ’03)*, pp. 42–51, Aachen, Germany, June 2003.

International Journal of Digital Multimedia Broadcasting

Special Issue on P2P Multimedia Social Networking and Communication Systems

Call for Papers

Social Networking has extremely impacted the communication habits of people on the internet. On the other hand, increasingly people carry their treasured multimedia content in small mobile devices like smart phones, PDAs, mp3/dvd players, memory sticks, cameras, and so forth. Tremendous interesting customized applications can be imagined with spontaneous social networking over mobile devices in the events/locations such as conferences, expositions, galleries, stadiums, and finally bars and restaurants. Enabling people to share their experiences, to communicate, and to have access to the comments of others without need to have internet access and with minimum required infrastructure is the main idea behind spontaneous social networking. Considering, auto-organization, scalability, and distributed architecture, P2P technology can play the key role in creating such kind of social networking.

The focus of this special issue is on the P2P spontaneous social networks. The limited resources on the mobile devices, the ad hoc wireless network instability, and the mobility of the nodes raise important challenges. The scalability of the feeding system, the reliability, data dissemination, and adaptation to the restrictions in ad hoc networks conditions can be considered as the main humps in the way. This special issue invites the original papers in the following topics, not limited to them:

- P2P communication systems
- P2P IP Telephony protocols
- P2P social networking techniques
- Multimedia social networking: theory and practice
- Social networking over delay tolerant networks (DTNs)
- P2P technologies and network coding
- Scalability of the P2P feeding and alert systems
- Adaptation of P2P overlays for low-mobility ad hoc networks
- P2P communication systems for vehicular systems
- Modeling of P2P communication systems
- Real experiments on P2P social networking and communication systems
- Tagging and indexing in P2P social networks
- Community-based computing
- P2P tracing and monitoring systems

Before submission authors should carefully read over the journal's Author Guidelines, which are located at http://www.hindawi.com/journals/ijdmb/guidelines.html. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/, according to the following timetable:

Manuscript Due	August 1, 2009
First Round of Reviews	November 1, 2009
Publication Date	February 1, 2010

Lead Guest Editor

Mehdi Mani, Network and Service Architecture Laboratory, Department of Wireless Networks and Multimedia Service, Institut TELECOM, Telecom & Management SudParis, 91011 Evry Cedex, France; mehdi.mani@it-sudparis.eu

Guest Editors

Noël Crespi, Department of Wireless Networks and Multimedia Service, Institut TELECOM, Telecom & Management SudParis, 91011 Evry Cedex, France; noel.crespi@it-sudparis.eu

Winston Seah, The Networking Protocols Department, Institut for Infocomm Research, Singapore 138632; winston@i2r.a-star.edu.sg

Golnaz Karbaschi, ASAP Reseach Group, INRIA, 91893 Orsay Cedex, France; golnaz.karbaschi@inria.fr

EURASIP Journal on Audio, Speech, and Music Processing

Special Issue on Scalable Audio-Content Analysis

Call for Papers

The amount of easily-accessible audio, either in the form of large collections of audio or audio-video recordings or in the form of streaming media, has increased exponentially in recent times. However, this audio is not standardized: much of it is noisy, recordings are frequently not clean, and most of it is not labeled. The audio content covers a large range of categories including sports, music and songs, speech, and natural sounds. There is, therefore, a need for algorithms that allow us make sense of these data, to store, process, categorize, summarize, identify, and retrieve them quickly and accurately.

In this special issue, we invite papers that present novel approaches to problems such as (but not limited to):

- Audio similarity
- Audio categorization
- Audio classification
- Indexing and retrieval
- Semantic tagging
- Audio event detection
- Summarization
- Mining

We are especially interested in work that addresses real-world issues such as:

- Scalable and efficient algorithms
- Audio analysis under noisy and real-world conditions
- Classification with uncertain labeling
- Invariance to recording conditions
- On-line and real-time analysis of audio
- Algorithms for very large audio databases

We encourage theoretical or application-oriented papers that highlight exploitation of such techniques in practical systems/products.

Before submission, authors should carefully read over the journal's Author Guidelines, which are located at http://www.hindawi.com/journals/asmp/guidelines.html. Authors should follow the EURASIP Journal on Audio, Speech, and Music Processing manuscript format described at the journal site http://www.hindawi.com/journals/asmp/. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/, according to the following timetable:

Manuscript Due	June 1, 2009
First Round of Reviews	September 1, 2009
Publication Date	December 1, 2009

Lead Guest Editor

Bhiksha Raj, Carnegie Mellon University, PA 15213, USA; bhiksha@cs.cmu.edu

Guest Editors

Paris Smaragdis, Advanced Technology Labs, Adobe Systems Inc. Newton, MA 02466, USA; paris@media.mit.edu

Malcolm Slaney, Yahoo! Research, Santa Clara, CA 95054; Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, CA 94305-8180, USA; malcolm@ieee.org

Chung-Hsien Wu, Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan; chwu@csie.ncku.edu.tw

Liming Chen, Department of Mathematics and Informatics, Ecole Centrale de Lyon University of Lyon, 69006 Lyon, France; liming.chen@ec-lyon.fr

Hyoung-Gook Kim, Intelligent Multimedia Signal Processing Lab, Kwangwoon University, Seoul 139-701, South Korea; hkim@kw.ac.kr

Hindawi Publishing Corporation
http://www.hindawi.com

EURASIP Journal on Wireless Communications and Networking

Special Issue on
Theoretical and Algorithmic Foundations of Wireless Ad Hoc and Sensor Networks

Call for Papers

This special issue is devoted to distributed algorithms and theoretical methods in the context of wireless ad hoc and sensor networks. Recent research in mobile ad hoc networks and wireless sensor networks raises a number of interesting, and difficult, theoretical and algorithmic issues. While much work has been done in protocol and system design, simulation, and experimental study for wireless ad hoc and sensor networks, the theoretical research, however, falls short of the expectation of the future networking deployment. The needs to push the theoretical research forward for a deeper understanding about wireless ad hoc and sensor networking and to foster cooperation among networking researchers and theoreticians establish the motivation behind this special issue.

The objective of this special issue is to gather recent advances in the areas of wireless ad hoc and sensor networks, with a focus on theoretical and algorithmic aspect. In particular, it will concentrate on distributed algorithms, randomized algorithms, analysis and modeling, optimizations, and theoretical methods in design and analysis of networking protocol (at link layer or network layer) for wireless ad hoc and sensor networks. Specific topics for this special issue dedicated to theoretical and algorithmic foundations include but are not limited to:

- Channel assignment and management
- Distributed and localized algorithms
- Dynamic and random networks
- Dynamic graph algorithms
- Energy conservation methods
- Localization and location tracking
- Mechanism design and game theory
- Modeling and complexity analysis
- Routing, multicast, and broadcast
- Scheduling and synchronization
- Throughput optimization and capacity

Before submission, authors should carefully read over the journal's Author Guidelines, which are located at http://www.hindawi.com/journals/wcn/guidelines.html. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/, according to the following timetable:

Manuscript Due	November 1, 2009
First Round of Reviews	February 1, 2010
Publication Date	May 1, 2010

Lead Guest Editor

Yu Wang, Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA; yu.wang@uncc.edu

Guest Editors

Xinbing Wang, Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai 200240, China; xwang8@sjtu.edu.cn

Ping Ji, Department of Mathematics and Computer Science, John Jay College of Criminal Justice, City University of New York, New York, NY 10019, USA; pji@jjay.cuny.edu

Peter Müller, IBM Zurich Research Laboratory, 8803 Rüschlikon, Switzerland; pmu@zurich.ibm.com

Xiaohua Jia, Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong; csjia@cityu.edu.hk

Athanasios Vasilakos, Department of Computer and Telecommunications Engineering, University of Western Macedonia, Greece; vasilako@ath.forthnet.gr