PROGRESSIVE VIEW-DEPENDENT TRANSMISSION OF 3D MODELS OVER LOSSY NETWORKS

B. Tarı, Y. Yemez, Ö. Özkasap, R. Civanlar

Computer Engineering Department, Koç University
Rumeli Feneri Yolu, 34450, Sarıyer, Istanbul, Turkey
{btari, yyemez, oozkasap, rcivanlar}@ku.edu.tr

ABSTRACT

In this paper, we present an interactive view-dependent technique for streaming progressively encoded 3D models over lossy networks. The 3D model data is represented and encoded with progressive octree particles. TCP/IP protocol assures reliable transmission but does not well suit to real-time interactive graphics streaming over lossy networks with limited bandwidth. Though being unreliable, UDP provides faster transmission. Thus we adapt a hybrid dynamic transmission scheme which uses TCP only to transmit some minimal information needed for the basic shape structure and which transmits the appearance attributes of visible surface points with UDP. A significant reduction in the amount of data transmitted to realize a possible interactive client-server scenario is achieved for graphics streaming over a lossy network with almost no noticeable degradation in the visual quality. This is thanks to the view-dependency and loss recovery support of the proposed technique. Simulation results demonstrate the efficiency of the interactive streaming technique and the visual quality achieved.

1. INTRODUCTION

As the Internet expands and processing capabilities of computers improves, demand is growing for high resolution 3D models from games to commercial websites in various distributed applications. While high resolution 3D models create desirable visual effect, problems arise quickly: a large amount of data must be stored, transmitted and rendered within a limited bandwidth and time interval. These criteria have been motivation for development of new representation and transmission schemes for typical 3D meshes in its early times. Single level compressed models, for example, decreased storage, hence also the end-to-end delay. Yet they do not provide something to be rendered until all data is downloaded. Another possibility is the use of flexible representations and interactive techniques which allow the user to choose the best compromise between fast data access and high quality visualization. Progressive view-dependent schemes seem to serve well for this purpose since an exact representation of the geometry is not always required for applications such as navigation or browsing with limited bandwidth transmission. Moreover, the need and capability for complexity may vary from one user to another. Although, progressive meshes (PM) [3] and compressed PM [4] methods, each of which defines a base mesh and refinement operations to reach higher level of resolution, can reduce the latency, they do not well scale by their nature to unreliable but faster transmission schemes in becoming fragile upon packet losses because of inter-packet dependency of refinement operations. The fragility can be avoided by deploying error-free TCP. However, TCP is undesirable because of its costly reliability mechanism. While a TCP sender can notice a lost, it will respond by keeping retransmitting this packet until being sure of its delivery and decreasing its transmission rate, all of which increase latency. Furthermore, TCP does not suit multicast-scenarios where the channel would suffer from retransmission requests of each single receiver. Another reason of TCP's being unsuitable in transmission of data of large amount arises in a network where a handheld host possibly connects via wireless link. TCP cannot differentiate the reason of the packet loss caused by usual bit errors of the wireless link from that of congestion and respond in both cases by decreasing the sending rate.

The disadvantages of totally reliable TCP have led to the idea of utilization unreliable but faster UDP for graphics applications. The hybrid transmission scheme presented in [5] proposes reliable transmission of data where needed and unreliable where loss can be tolerated. The loss resilient transmission presented in [6], on the other hand, explores optimized unequal error protection through FEC (Forward Error Correction) deployment where level of protection over a packet is determined according to the distortion caused by the loss of this packet.

In the context of progressive transmission over lossy channels much of the work focuses on the effect and recovery of the loss of triangle based representations. However, point based representations like octree particles [1] and QSplat representation [2] are more appropriate for transmission over lossy networks, as they take the advantage of the fact that connectivity information can be dropped for both of the cases. The benefit of using a connectivity omitting representation has its advantage by diminishing a significant amount of the total data to be transmitted. More importantly, the need to update the connectivity information with refinement operations for regular progressive mesh schemes is the main source of fragility over lossy networks.

The existing schemes for 3D streaming can be classified depending on whether they consider the following criteria: View dependency, loss recovery, level of fragility of the model in lossy networks. Rusinkiewicz et al. [2] propose a view dependent streaming version of their point-based representation Qsplat. Their system eliminates connectivity information. However, since the transmission scheme is fixed to TCP, loss of data is not explored. Al-Regib et al. [6] apply unequal error protection over CPM [4]. View dependency and loss recovery lack in their work. Nevertheless, the transmission stops when the loss exceeds the allowed limit. Chen et al. [5] propose a hybrid TCP/UDP transmission scheme. But they use PM [3] which is connectivity preserving and the system is very fragile to packet losses and does not address view-dependency. A very recent work [7] presents an interactive streaming technique for view-dependent progressive mesh transmission, but does not address the error-resiliency problem.

In this paper, we present a view-dependent streaming version of the progressive point-based scheme (octree particles) introduced by Yemez et al. [1]. We deploy a hybrid transmission scheme: The basic 3D octree structure belonging to the requested LOD (level of detail) is transmitted using a reliable transmission protocol (TCP), whereas extra geometric precision information and appearance attributes such as colors and normals, belonging to visible parts of the model, are unreliably sent. No extra information is needed for the visibility which is computed on-the-fly. Precision and attributes are incrementally transmitted as the view angle and/or level of detail changes. Since a reliable transmission is used for the basic 3D structure, the system never breaks down because of possible losses that might occur during unreliable transmission of precision and attributes. Moreover, the loss of appearance attributes can be recovered up to a certain level via neighborhood approximation. To best of our knowledge, our scheme is the only interactive 3D streaming scheme in the literature, that is progressive, view-dependent and loss-resilient at the same time.

This paper is organized as follows: In Section 2 we define the problem, that is the interactive scenario that we consider between client and server. Section 3 briefly reviews the octree particle scheme that we use for 3D representation In Sections 4 and 5, the hybrid TCP-UDP transmission scheme and view-dependent streaming mechanism are described respectively. Section 6 addresses our packaging strategy and loss recovery. In Section 7 we present our experimental results and finally give some concluding remarks in Section 8.

2. CLIENT-SERVER INTERACTION

We consider a client-server graphics streaming scenario. The client initiates a demand to visualize a 3D model stored in the server. The server initially sends a coarse version of the detailed model from a predefined view-point, that is displayed on the client side. The client can then send requests to the server in order to display the model from different view-points and at different levels of detail. The server is thus charged to respond to two kinds of demand: The user may desire more detail or change its point of view. In both cases, the server determines what information to be sent and which parts of the model have been uncovered after the view change, and prepares relevant packets in an agreed way so that the decoder on the client side can understand the content of each packet without specific identifiers. The client side resolves the content of each packet, applies loss recovery if needed and renders the resulting model. The client-server 3D streaming model is depicted in Fig.1.

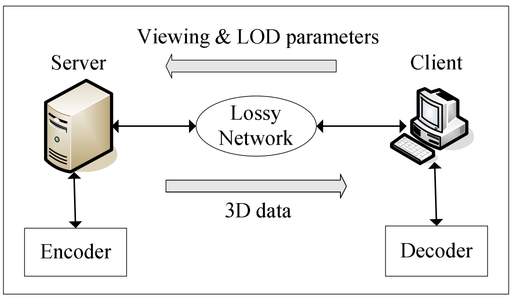

Fig 1: Client –server interactive 3D streaming model

3. OCTREE PARTICLES

Our 3D representation is based on the octree particles scheme of Yemez et al. [1], which can be regarded as a progressive point-based modeling technique. Here, we will briefly review the octree particles scheme; the reader should refer to [1] for a detailed description. In this scheme, the object surface is first partitioned into cubic voxels by the hierarchical octree structure. This partitioning produces a sequence of octree surfaces, S_0, S_1,..., S_R, which voxelize the surface geometry at R different levels of detail, where R is the leaf node level of the octree. In order to represent the object surface with a more precise shape as well as other surface attributes, each octree surface node is associated to a particle. A particle is defined as a surface point with attributes such as geometric position, color and normal. The surface particles yield compact and space efficient encoding of overall object appearance, and when arranged and encoded in an appropriate order, without need for any extra information, they can progressively be decoded to reconstruct the octree surfaces S_0, S_1,..., S_R, and the related surface attributes. These LOD surfaces can then be rendered by the viewer using either a splatting or a fast direct triangulation technique.

Color and normal attributes of an intermediate-level octree node are determined by averaging the attributes of its children starting from the leaf nodes, whereas its geometric position is determined with respect to one of its child nodes. The highest level of detail R may vary from one object to another and mainly depends on the shape complexity. When the object model is visualized at an intermediate level r, the precision of the geometric position is normally set to be higher than r to improve rendering quality.

4. HYBRID TRANSMISSION

Decoding of the octree particles representation is an incremental process. Each particle associated to an octree node at an intermediate level of detail is encoded with respect to its parent. Thus if the whole model is transmitted unreliably over a lossy network, the decoding process breaks down. In order to avoid this, the information encoded in octree particles representation is decomposed into two components. The first one is the basic octree structure that partitions the surface at R levels of detail and can be encoded by 8 bits per parent to specify statuses of its child nodes. Since this *structural information* is not tolerant to any loss, it must be transmitted using a reliable transmission protocol along with an additional surface orientation bit per node. The orientation bit is essential for correct rendering on the client side and is also necessary to perform visibility tests for a robust view-dependent interaction.

The second component of the octree particles representation, that is all the remaining information, is tolerant to losses, and can thus be transmitted unreliably. Geometric precision beyond the octree resolution, colors and normals are all in this category. Suppose that each surface point has a geometric precision of N bits. Then the octree particles representation contains per leaf node $3(N–R)$ bits of post precision for x, y and z coordinates. We will refer to these bits as *post-structural information*. Normally, colors or normals are each encoded by 24 bits per octree node, but the actual value is less than this since for instance the color of one of the 8 siblings can be recovered from the color of its parent and from those of the other siblings with a truncation error. We will refer to color and normal as *attribute information*.

To summarize, the structural information is transmitted by using TCP whereas for post-structural and attribute information, an unreliable transmission protocol such as UDP is employed.

5. VIEW-DEPENDENT STREAMING

In interactive view-dependent transmission schemes, ideally, only the portion of the 3D data, that is relevant to visible parts of the model is to be sent to the client. However, most of the existing LOD representations do not fully support this requirement due to the specific intrinsic characteristics of each modeling technique. Our approach partially meets this requirement since the structural information, which is essential to represent the surface, is sent regardless of the view point. Attribute and post-structural information transmitted can easily be adapted to the current view.

To clarify our view-dependent mechanism, suppose that the current level of detail requested by the client is r and the current view angle is θ. In this case, the whole structural information, visible from θ or not, at this level of detail is to be transmitted. The structural information at level r specifies only r bits of geometric precision for each coordinate axis, which is usually unsatisfactory for smooth visualization, leading to a jaggy surface. Some extra precision is needed and thus transmitted (usually $r+2$ precision is sufficient) for only visible parts of the model along with visible attribute information. If the client requests a view change by $\Delta\theta$ at the same level of detail, the octree particles falling onto the parts of the model uncovered after the view change are determined by the encoder and transmitted. If the client requests further refinement, the incremental structural information from r to $r+1$ for the whole model is transmitted along with the attribute information and extra precision corresponding to the view increment $\Delta\theta$. When the maximum level R of detail is reached and the client is content with the current view angle, some further refinement can still be requested, which is provided by the server if available, transmitting post-structural information, that is $(N-R)$ bits of precision beyond octree resolution for each coordinate axis.

As it is clear from the description above, view-dependent streaming needs to determine which of the octree nodes become visible each time the level of detail or view angle changes. The visibility information for each octree node can dynamically be updated during interactive transmission. This is achieved by on-the-fly computation of the normal vector for each octree node using the structural information and precision available at the current level of detail. A node is visible if the angle between its normal vector and the view direction is less than 90 degrees. Normal estimation relies only on the information available on the client side since the encoder and decoder must agree on which nodes are visible and to be transmitted. Recall that extra geometric precision bits $(r+2)$ are available only for currently visible parts at level r. The lack of precision on the invisible part of the surface may sometimes yield incorrect estimation of visibility for the uncovered nodes. Fortunately, correct visibility information can be recovered by iterating the visibility test, each time updating the normal estimates with incoming extra precision bits. The encoding/decoding procedure is depicted in Fig.2.

6. PACKETIZATION AND LOSS RECOVERY

Since congestion and thus packet loss is proportional to the number of transmitted bits in the network, we choose not to allow any identifiers within packets. This means that the encoder and the decoder must implicitly agree on the content of a packet. Thus during interactive transmission, both the encoder and the decoder dynamically build, hold and update the basic octree structure and its visibility status by using the reliable structural information. The content of the packets streamed over the lossy network is dynamically encoded and decoded by an octree traversal each time a new request is activated by the client.

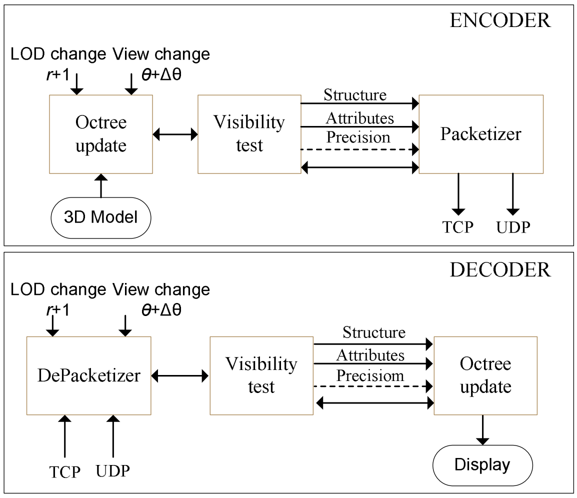

Fig 2: Encoder/decoder block diagram

The structural, attribute, precision and post-structural information bits are each encoded separately within independent packets by employing the appropriate transmission protocol. Since attribute and post-structural information is subject to loss, they are encoded so as to distribute relevant information of neighboring nodes on the model surface into different packets. In this way, the effect of a packet loss is spread over the surface and does not concentrate on some particular region. Table 1 displays different types of streamed model information, their delivery schemes and bitloads per surface point, i.e. per leaf node [1]. Clearly, the portion of the 3D data sent by UDP is significantly larger than the data sent by TCP.

Type	Protocol	View-dependency	Bits
Octree structure	TCP	No	~5
Orientation	TCP	No	~1.5
Color	UDP	Yes	~29
Normal	UDP	Yes	~29
Post-structure	UDP	Yes	$3\cdot(N-R)$

Table 1: Packet types, delivery schemes and approximate bitloads per surface point needed for progessive representation; R is the maximum octree level and N is the maximum geometric precision available for a surface point.

At any instant of interactive streaming, the decoder on the client side, having constructed the same octree structure and visibility information as the encoder, knows exactly how many UDP packets have been sent and to what they correspond. If the packet sequence numbers are successive but the number of packets is less than expected, then this means that the last packets have been lost. Otherwise the missing packets are deducted by looking at the packet sequence numbers. In case an attribute packet is found to be lost, the attribute of every node addressed in the lost packet can be approximated by the attribute of its parent. Since the attribute of a parent is the average of its child nodes, this approximation strategy serves as an effective loss recovery method, preventing the occurrence of unpleasant visual artifacts. No loss recovery is needed for post-structural information since the system is not fragile against precision losses beyond octree resolution. Post-structural information loss may become noticeable by the viewer only in the case of zooming.

7. RESULTS

The proposed interactive client-server streaming technique has been simulated on a high resolution 3D model (Anyi statuette) by applying a discrete random packet loss with error rate of 10%. This simulation drops one of every ten packets in UDP channel. The model of the Anyi statuette contains approximately 90,000 surface points along with color and normal information, and 16-bit geometric precision.

Fig.3 demonstrates the loss recovery mechanism for color attribute information at incremental levels of detail. As it can be observed, the loss is spread uniformly over the surface and hardly noticeable thanks to distributed packetization and the recovery strategy. Fig.4 demonstrates a possible streaming scenario between server and client. The user starts with downloading and rendering an initial coarse model viewed from a predefined angle. Then she interactively changes view angle and can request further detail from certain view points. When she becomes content with a view point, the model is display at maximum resolution along with available post-structural information. The effect of losses is hardly noticeable. In Fig. 5, some snapshots from invisible parts of the model from various view angles are displayed to demonstrate the view-dependency of the proposed technique. The amount of the data transmitted during this interactive streaming scenario is compared in Table 2 with the total size of the Anyi statuette at the highest resolution with post-structural and color attribute information, where we observe a significant reduction and thus gain in the exploited bandwidth with the use of the proposed scheme.

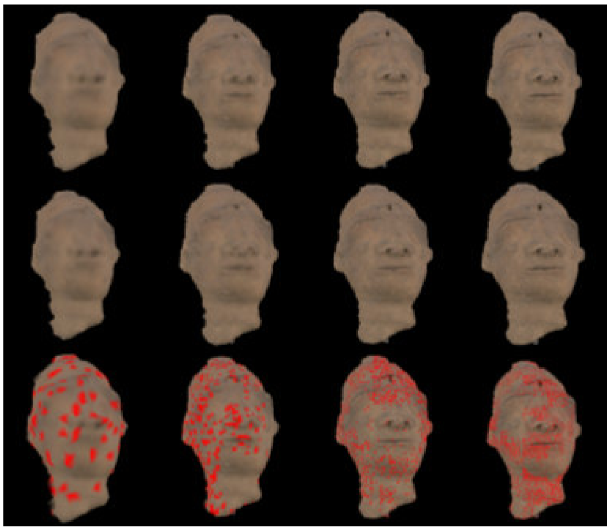

Fig 3: Loss recovery for color at incremental levels of detail with $r = 6,7,8,9$. (First row) lossless streaming; (second row) recovered from 10% loss; (last row) distribution of losses over the surface, red labels indicate places where color is either lost or decoded incorrectly.

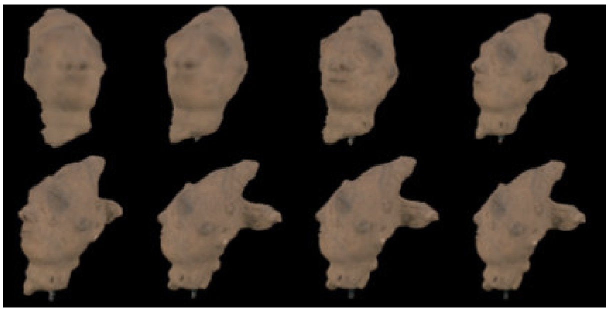

Fig 4: Streaming scenario: (First row) a) initialization with coarse model at $r = 6$, b) after rotation with $\Delta\theta = 30°$, c) $r = 7$, d) after rotation with $\Delta\theta = 30°$; (second row) e) $r = 8$, f) after rotation with $\Delta\theta = 30°$, g) $r = 9$, h) after applying all available geometric precision.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a view-dependent progressive hybrid 3D transmission scheme over lossy networks by using the octree particles representation [1]. The proposed scheme sends the basic structural information of the 3D model, which is not tolerant to loss, by using a reliable transmission protocol (TCP), whereas for the appearance attributes and post-structural information, an unreliable delivery scheme (UDP) can be employed. A significant reduction in the amount of data transmitted to realize a possible interactive client-server streaming scenario is achieved over a lossy network thanks to the view-dependency and loss recovery support with almost no noticeable degradation in the visual quality.

Future work will involve 1) application of unequal error protection on the packets of different types, which are sent unreliably, 2) using a more realistic loss simulation, rather than discrete random loss, by placing a realistic packet loser between the server and the client.

Fig 5: View dependent transmission. Reverse angle snapshots for a), b), d) and f) of Fig. 4.

Type	Total	Interactive
Structure and Orientation	177738	177738
Color, Normal and Post-Structure Precision	916636	562888
Total of delivered data in bytes	1094374	740626

Table 2: Comparison of delivered data in bytes with the proposed interactive transmission scheme to realize the scenario given in Fig. 4 to the total size of the Anyi statuette.

REFERENCES

[1] Y. Yemez and F. Schmitt, “Multilevel Representation and Transmission of Real Objects with Progressive Octree Particles”, *IEEE Trans. on Visual. and Comp. Graphics*, Vol. 9, No. 3, 2003.

[2] S. Rusinkiewicz and M. Levoy, “Streaming QSplat: A Viewer for Networked Visualization of Large, Dense Models”, *Proc. 2001 Symposium on Interactive 3D Graphics*, 2001.

[3] H. Hoppe, “Progressive Meshes,” *Proc. ACM SIGGRAPH*, pp. 99-108, 1996.

[4] R. Pajarola, and J. Rossignac, “Compressed progressive meshes,” *IEEE Trans. on Visual. and Comp. Graphics*, Vol. 6, No. 1, pp. 79-92, 2000.

[5] Z. Chen, B. Bodenheimer, and J. F. Barnes, “Robust Transmission of 3D Geometry Over Lossy Networks”, *Proceedings of the 8th International Conference on 3D Web Technology*, pp. 161-173, St. Malo, France, 2003.

[6] G. Al-Regib and Y. Altunbasak, “An unequal error protection method for packet loss resilient 3-d mesh transmission,” In *Proceedings of INFOCOM* 2002 (2002).

[7] S. Yang, C. Kim, and C.-C. J. Kuo, “A Progressive View-Dependent Technique for Interactive 3-D Mesh Transmission”, *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 14, No. 11, pp. 1249-1264, 2004.