

3D Model Streaming based on a JPEG 2000 Image

Nein-Hsiung Lin, Ting-Hao Huang, and Bing-Yu Chen, *Member, IEEE*
National Taiwan University

Abstract-- For PC and even mobile devices, video and image streaming technologies, such as H.264 and JPEG/JPEG 2000, are already mature. However, the 3D model streaming technology is still far from practical use. Therefore, we wonder if 3D model streaming can directly benefit from current image and video streaming technologies. Hence, in this paper, we propose a mesh streaming method based on geometry image [2] to represent a 3D model or a 3D scene and integrate it into an existed client-server multimedia streaming server. In this method, the mesh data of a 3D model is first converted into a JPEG 2000 (J2K) [3] image. Based on the JPEG 2000 streaming technique, the mesh data can then be transmitted over the Internet as a mesh streaming. Moreover, since this method is based on JPEG 2000 standard, our system is much suitable to be integrated into any existed image and video streaming system.

I. INTRODUCTION

Recently, 3D graphics over the Internet has attracted a lot of attention, such as web-based virtual shopping malls and on-line 3D games. For supporting this, the demand of transmitting 3D models increased significantly. Being able to view a 3D model or a 3D scene composed of many sophisticated 3D models over the Internet is one of the goals of VRML (Virtual Reality Modeling Language) and X3D (eXtensible 3D). However, due to the increase of model complexity and file size, even with the increase of network bandwidth, to download the 3D models would still take a lot of time. To reduce the waiting time for downloading the 3D models, mesh streaming mechanism must be available as what has been done in the video and image streaming.

Therefore, in this paper, we propose a new mesh streaming method by utilizing the benefits of JPEG 2000. This method takes the advantage of the fact that a 3D model can be represented by a geometry image [2]. This could reduce the problem of 3D mesh streaming and transfer it to 2D image streaming. There are also many 2D image compression methods that can be used to further downsize the file of the geometry image. This paper made use of the JPEG 2000 [3] compression due to some of its good characteristics, such as ROI (Region Of Interest), progressive compression, multiple components, etc. Moreover, since this method is based on JPEG 2000, which is a famous standard, our system is much suitable to be integrated into any existed image and video streaming system.

Based on our mesh streaming method, during the downloading process, the user can first obtain an approximate shape and then the 3D model will become clearer when more data is received. Besides the progressive transmission, the

view-dependent issue is also taken into account. Hence, the most significant part of the 3D model or the part faced to the user will be refined earlier.

II. ALGORITHM

Our 3D model streaming method first produces the geometry image of a 3D model. A geometry image is an extension of surface parameterization; more specifically, it uses an image's R, G, B values to represent the model's attributes, such as vertex coordinates, normal maps, texture coordinates, etc. Surface parameterization, on the other hand, is to find an one-to-one mapping function $F: \mathbb{R}^3 \leftrightarrow \mathbb{R}^2$, such that

$$F(p_i) = q_i, \text{ where } \begin{cases} p_i = (x_i, y_i, z_i) \\ q_i = (u_i, v_i) \end{cases}.$$

Thus, while processing a model, we first use the cut method in [2] to find the proper boundary of the model. Then, we use Floater's surface parameterization method [1] to flatten the model which has been cut to be a surface of 2-manifold with boundary. Hence, the model is flattened to a 2D surface and the boundary of the model is mapped to a square. We then resample the grid points in the image and use interpolation to calculate the attributes, such as vertex coordinates, normal maps, and texture coordinates. Finally, we normalize the attributes to get the corresponding R, G, B values.

The geometry image is not the actual image that we transmit over the Internet, due to its file size and the fact that we have to transmit a number of images to recover all of the attributes. JPEG 2000 compression method was employed to address this problem, which is chosen for the reasons below:

- JPEG image is the most used image compression format on the Internet, and so is JPEG 2000.
- JPEG 2000 has a lossless compression mode.
- JPEG 2000 has greater compression rate compared to the older methods.
- JPEG2000 supports multiple layer compression, thus we can compress the data for all attributes into one image.
- JPEG 2000 supports progressive compression / decompression, which allowed us to achieve progressing transmission over the Internet.
- Arbitrary image block can be retrieved from a JPEG2000 image, which is very suitable for view dependent transmission.

III. CLIENT-SERVER ARCHITECTURE

Fig. 1 shows the concept of the transmission under our client-server architecture. The further explanations are as the following:

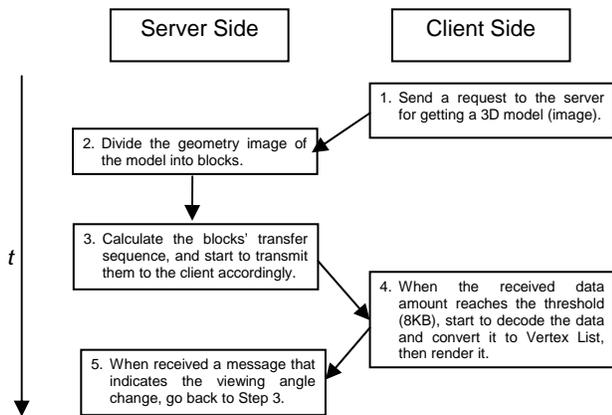


Fig. 1. The concept of the client-server transmission.

1. When the client requests the server for getting a certain 3D model, together with the request, the initial viewing information is also sent to the server. The viewing information is a 4x4 transform matrix. From this matrix, we can calculate the vertex that is closest to the current viewpoint by

$$z'_i = \text{rotation}[2] \times x_i + \text{rotation}[6] \times y_i + \text{rotation}[10] \times z_i \cdot$$

2. When the server receives the transmission request and the viewing information, the server first divides the geometry image of the 3D model into blocks. The default block size is 8x8 according to the block size used in DCT (Discrete Cosine Transform) of JPEG, thus for the image with resolution of 32x32, it is divided into 4x4 blocks.
3. With the number of blocks and the viewing information, the server can calculate which block is closest to the user's view and then decide the first block to be sent.
4. After deciding the first block to be sent, we have to decide what are succeeding blocks. To transmit the blocks as near as possible to the first block, we send the blocks in a swirl fashion. The first one being the eye of the swirl, the next block in line would be the one at its immediate top, then the one next to the second in a counter-clockwise fashion. After obtaining the sequence, we can place the blocks into the sending buffer according to this sequence. In the process of transmission, the server will keep track of the block indices, so that when the viewpoint changes, there is no need to re-send the blocks that has been sent already.
5. After the client receives the block, it can start decoding and placing the information into an array prepared beforehand according to the block index. Squares with incomplete vertices will be set aside first. With this, we can decode while transmission and achieve the mesh streaming.

When the viewpoint changes, the client will resend the viewing information to the server as described in Step 1. When the server receives the message, it will stop to transmission of the current blocks it is transmitting, then proceed to Step 3 to re-compute the new transmission sequence and start to transmit again.

IV. RESULT

To show the result, we use a Beethoven model. The original file size is 314KB in .obj file format and its 256x256 geome-

try image (J2K) file size is 105KB. Fig. 2 shows the multi-resolution result. With the multi-layer JPEG 2000 image, we can see the model being rendered in greater details as more data received by the client as time goes.



Fig. 2. Multi-resolution result.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a brand new 3D model streaming method which used geometry image to store all information of a 3D model. Since the geometry image is encoded as a JPEG 2000 image, we can compress the geometry image without data loss (lossless mode) or compress it in high compression rate with a little bit data loss (lossy mode). With JPEG 2000's support for progressive compression / decompression, we can progressively transmit and render a 3D model to reduce the user waiting time.

However, even the lossless mode of JPEG 2000 can compress the data without losing quality; there are still some inherent problems. To convert the 3D model to a geometry image, we have to first cut the model to find the boundary and this actually may cause data loss. Moreover, when using Floater's method for surface parameterization, there is a tendency to loss the information at sharp regions of a model. Hence, the surface parameterization method should be enhanced to lessen the data loss.

REFERENCE

- [1] M. Floater, "Parametrization and smooth approximation of surface triangulations," *Computer-Aided Geometric Design*, vol. 14, no. 3, 1997, pp. 231-250.
- [2] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," *ACM Transactions on Graphics (SIGGRAPH 2002 Conference Proceedings)*, vol. 21, no. 3, 2002, pp. 355-361.
- [3] JPEG 2000 Part 1 Final Committee Draft Version 1.0, ISO/IEC FCD15444-1:2000, 2000.