PROGRESSIVE CODING OF 3D TEXTURED GRAPHIC MODEL VIA JOINT MESH-TEXTURE OPTIMIZATION

Sheng Yang, Meiyin Shen and C.-C. Jay Kuo

Integrated Media Systems Center and Department of Electrical Engineering
University of Southern California, Los Angeles, California 90089-2564

ABSTRACT

Progressive coding of 3D textured graphic models using a joint mesh-texture optimization technique is investigated in this work. The mesh and texture data of a model are first fed into their respective compression modules to result in a series of levels of details. Then, for a given viewpoint, a rate-distortion surface is generated using these mesh and texture data. Afterwards, an optimal path over the rate-distortion surface is determined by the steepest descent algorithm. To achieve progressive transmission, the mesh and texture are transmitted in a certain ratio along the optimal path to provide the best visual quality for the given viewpoint. For further generalization, a layered sampling algorithm is proposed to deal with an arbitrary viewing angle. The performance of the proposed joint mesh-texture progressive coding algorithm is demonstrated by experimental results.

1. INTRODUCTION

Progressive coding and transmission of 3D graphic models has attracted a lot of attention recently in academia and industry for applications such as architectural presentations, virtual marketing, and network gaming, etc. In these applications, it is desirable to progressively encode and transmit both mesh data and associated textures to reduce the lengthy transmission and rendering time. For most existing graphic coding research, the focus has been on mesh compression. For a textured graphic model, we may treat the texture and the mesh as two separate entities and compress them independently and progressively. However, little research has been done on the multiplexing of these two bit streams to achieve the best visual quality under the progressive transmission environment.

This is a challenging question, since there is no simple metric to measure the distortion of compressed textured graphic models. For an intermediate mesh of a compressed graphic model, the distortion comes from three sources: the geometric error, the surface warping effect, and the texture distortion. The geometric error is introduced because of the inaccuracy of vertex positions. Surface warping is resulted from the texture mapping deviation due to the error of texture coordinates. Texture distortion is the distortion of a textured image due to its lossy representation. Note that most previous work on progressive graphic coding [1, 2] dealt with the mesh data alone, where only the geometric error is relevant. In this research, we propose a general rate-distortion model that takes both mesh and texture data into account so that the bit allocation between them can be optimized to achieve the best display quality at any truncated point of the coded bit stream.

There are three major contributions of this research. First, we provide a new approach to characterize the relationship between the bit rate and the display quality of a textured graphic model. Second, an optimal strategy is proposed to multiplex mesh and texture data into one single bit stream in order to present the best display quality subject to the bandwidth limitation. Third, a layered sampling method is proposed for image rendering when viewers look at the model from different viewpoints.

The rest of the paper is organized as follows. Sec. 2 describes the process of calculating the rate-distortion surface and the method of multiplexing mesh and texture data. Sec. 3 explains the layered sampling method. Experimental results are given in Sec. 4 to demonstrate the performance of the proposed algorithm.

2. RATE-DISTORTION OPTIMIZATION FOR FIXED VIEWPOINT

The proper multiplexing of mesh and texture data into one single bit stream is important to the delivery of textured graphic models. To answer this question, we need to derive the rate-distortion relationship. Since we can allocate bit rates to mesh coding and texture coding with a certain ratio, the bit rate allocation defines a plane and the distortion is a function (or surface) defined on this bit-rate plane.

In this work, we use the edge collapse/vertex split method [3] to encode the mesh data and the wavelet-based image compression standard, JPEG2000 [4], to encode the texture data. It is worthwhile to point out that our rate-distortion study is not confined to these specific codecs. Our objective is to build a generic framework applicable to all possible progressive codecs. The specific codecs adopted only serve as a concrete example for ease of discussion.

After compression, the rate-distortion functions for mesh and texture data can be written as

$$\begin{aligned} D_m &= f_m(R_m), \\ D_t &= f_t(R_t), \end{aligned}$$

where D_m and D_t are mesh and texture distortion measures, and R_m and R_t are mesh and texture coding rates, respectively. Typically, the mesh distortion is defined in the 3D mesh space, while the texture distortion is defined in the 2D image space. Therefore, it is difficult to define a unifying metric to describe the distortion of a textured graphic model.

Furthermore, we assume that the mesh and the texture objects are represented by m and n layers, respectively, as

$$\begin{array}{ccccccc} M_0 & \rightarrow & M_1 & \rightarrow & \cdots & \rightarrow & M_{m-1}, \\ T_0 & \rightarrow & T_1 & \rightarrow & \cdots & \rightarrow & T_{n-1}. \end{array}$$

Then, there exist $m \times n$ joint mesh and texture representations:

$$\begin{array}{cccc} M_0T_0 & M_1T_0 & \cdots & M_{m-1}T_0 \\ M_0T_1 & M_1T_1 & \cdots & M_{m-1}T_1 \\ \vdots & \vdots & \ddots & \vdots \\ M_0T_{n-1} & M_1T_{n-1} & \cdots & M_{m-1}T_{n-1} \end{array}$$

To address the distortion measure issue, we first consider the case, in which a viewer looks at the model from a fixed viewpoint. Under this scenario, to measure the joint distortion associated with M_iT_j with $0 \leq i < m$ and $0 \leq j < n$, we may render the model from the given viewpoint, and compare the difference of this image with that rendered in the full resolution of both mesh and texture, *i.e.* $M_{m-1}T_{n-1}$. Then, the distortion is actually defined on the 2D image domain, and the commonly used distortion metric such as the Mean Square Error (MSE) is adopted to calculate the joint distortion. For any combination M_iT_j, we can obtain such a distortion measure. Finally, distortions for all $m \times n$ combinations form a surface, called the rate-distortion surface. One example is illustrated in Fig. 1.

The rate-distortion surface can be expressed as

$$D_j = f_j(R_m, R_t),$$

where D_j is the joint distortion defined in the rendered image domain, R_m and R_t are bit rates for the mesh coding and the texture coding, respectively. To maximize the visual quality, we seek a path over the rate-distortion surface which makes the joint distortion decrease the fastest.

In practice, we have the integer constraint on the mesh layer index i and the texture layer index j as formulated above. Thus, if we start from representation M_iT_j, we only have two choices for the next higher rate, *i.e.* $M_{i+1}T_j$ and M_iT_{j+1}. Thus, at each stage, we simply compare the rate-distortion slope for these two cases and choose the one that has the steeper slope. The process is repeated until the mesh and texture data have been completely delivered. The result is rather accurate if the gird is small enough.

The above algorithm works well due to the monotonicity of the rate-distortion surface in most regions. However, it is worthwhile to mention that, when the mesh is rendered in a very low resolution, the joint distortion will sometimes increase with the refinement of texture due to the fact that the large texture coordinate error results in severe texture deviation. In other words, the steepest descent method may not guarantee the global optimal path in the non-convex region. However, this can be treated as a pathological case, and it only occurs at very low bit rates. On the other hand, the refinement of mesh will always decrease the joint distortion. Another way to exclude the pathological case is to demand the base rate for the simplest textured graphic model M_0T_0 to be greater than a reasonable threshold so that the rate-distortion surface possesses the monotonic decreasing property afterwards for larger values of i and j.

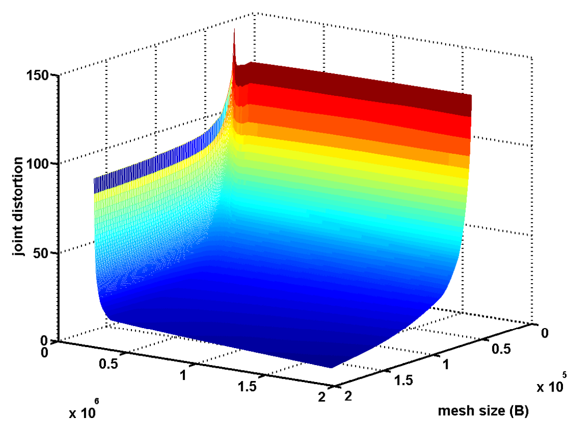


Fig. 1. The rate-distortion surface.

3. LAYERED SAMPLING

With the method described in Sec. 2, we have successfully solved the problem of joint texture-mesh progressive coding when the viewer looks at the model from a fixed viewpoint. However, usually the viewer will move around to observe the model from different viewpoints. In this section, we propose the layered sampling algorithm to address this problem. Due to the space limitation, we only discuss the scenario when the viewing distance between the viewer and the model remains unchanged. This result can be generalized to the case of varying distances, and will be reported later.

As a result of the constant viewing distance, the set of all viewpoints forms a sphere, which is called the viewing sphere. The idea of layered sampling is to approximate the viewing sphere by a progressive triangle mesh, called the viewing mesh. We first assign the four vertices of a regular tetrahedron internally tangent with the sphere as the first layer of the viewing mesh. Then, we continue subdividing the viewing mesh until it is fine enough. The process is similar to mesh subdivision. The viewing mesh construction is to iteratively insert new viewpoints so that the rate-distortion surface of arbitrary viewpoint can be predicted from its nearby viewpoints within a certain error bound.

Given three vertices vp_0, vp_1, and vp_2 of a triangle in the viewing mesh, we denote their rate-distortion surfaces as:

$$\begin{aligned} D_{j0} &= f_{j0}(R_m, R_t), \\ D_{j1} &= f_{j1}(R_m, R_t), \\ D_{j2} &= f_{j2}(R_m, R_t). \end{aligned}$$

For any viewpoint vp covered by the triangle, its actual rate-distortion surface is denoted by D_j. It can be approximated using the linear interpolation of results observed from vp_0, vp_1, and vp_2:

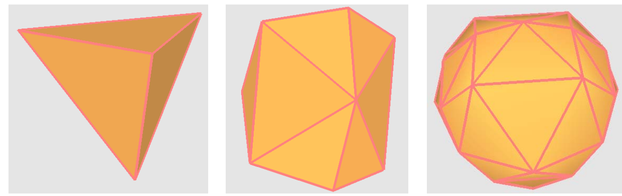
$$\begin{aligned} D'_j = (&\|vp - vp_0\| \cdot D_{j0} + \|vp - vp_1\| \cdot D_{j1} + \\ &\|vp - vp_2\| \cdot D_{j2})/(\|vp - vp_0\| + \|vp - vp_1\| \\ &+\|vp - vp_2\|), \end{aligned}$$

where $\|\cdot\|$ denotes the Euclidean norm in the 3D space. The approximation error can be calculated as

$$e = \int\int [D_j(R_m, R_t) - D'_j(R_m, R_t)]^2 dR_m dR_t.$$

We can continue to do the refinement until the maximum approximation error, which occurs at the barycenter of the triangle, is less than a given threshold.

To reduce the approximation error as much as possible, the following requirements of viewing mesh construction should be met. First, we should let triangles in each layer be approximately of the same size and a similar shape. Second, we should let the sides of the triangle be approximately of the same length. Thus, we adopt the viewing mesh subdivision method. That is, we first find the midpoints of each edge and extend them to intersect with the surface of the viewing sphere. The points of intersection are considered as new viewpoints. Then, we subdivide one triangle into four by connecting the newly inserted viewpoints. The first three layers are shown in Fig. 2. To measure the quality of subdivision, we calculate the ratio of the area of the biggest triangle to that of the smallest triangle in one layer. The ideal case is that every triangle has the same size. By our method, the ratios of the three layers are 1.00, 2.08, and 4.26, respectively.



(a) (b) (c)

Fig. 2. Illustration of the Viewing mesh: (a) the first layer, (b) the second layer and (c) the third layer.

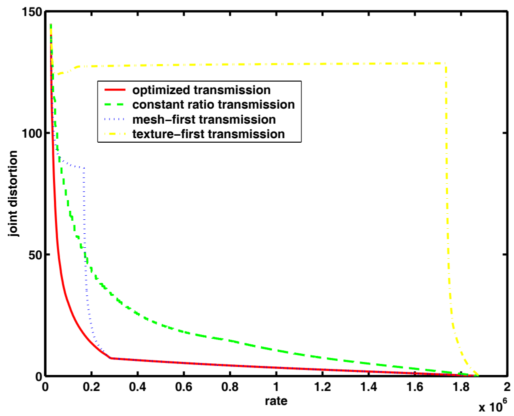
4. EXPERIMENTAL RESULTS

In this section, we test the proposed joint coding algorithm in two stages. First, we test if the rate-distortion surface can correctly reflect the relationship between the rate and the display quality of a textured graphic model and give the best transmission strategy. Second, we test if the layered sampling method works well for the case when the viewer looks at the model from different viewpoints.

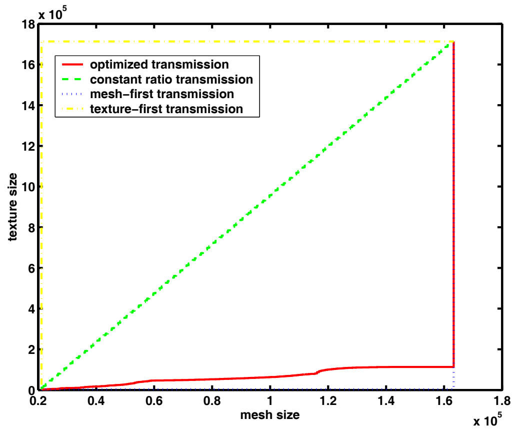
To demonstrate the performance of the proposed optimized joint coding algorithm, we compare it with the following three coding strategies: (1) the constant ratio scheme - the mesh and texture data are transmitted according to a constant ratio, which is determined by the total sizes of the mesh and the texture data; (2) the mesh-first scheme - the mesh data are transmitted first with the minimum amount of the texture data and, then, texture refinement data are transmitted after all mesh data have been completely delivered; (3) the texture-first scheme - the texture data are transmitted first with the minimum amount of the mesh data and, then, mesh refinement data are transmitted after all texture data have been completely delivered.

Let us take the model 'Bunny' as an example. The bit allocation results and the joint rate-distortion curves of these four coding strategies are illustrated in Fig. 3. At the beginning, the joint distortion of the mesh-first curve decreases dramatically, much faster than texture-first and the constant ratio schemes. Then, the decreasing speed is slowed down, since the contribution of mesh data becomes smaller. During this stage, the constant ratio scheme performs better than the mesh-first and the texture-first schemes. The mesh-first scheme drops again after starting to transmit texture data. The performance of the above three strategies varies according to the viewpoint. However, the proposed algorithm always outperforms the other three methods for any viewpoint over the whole transmission period.

As mentioned above, when the mesh resolution is very low, the joint distortion may sometimes increase with the refinement of the texture. This can be seen from the texture-first curve. Therefore, when the textured model is multiplexed using the constant ratio or the texture-first strategies,

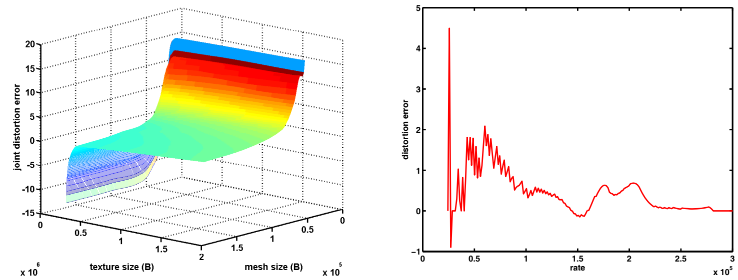


(a) The rate-distortion curve

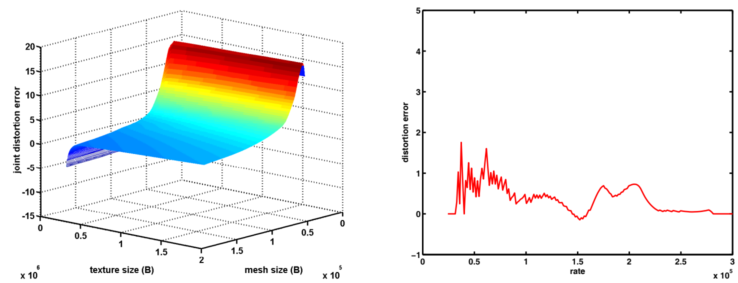


(b) Bit allocation

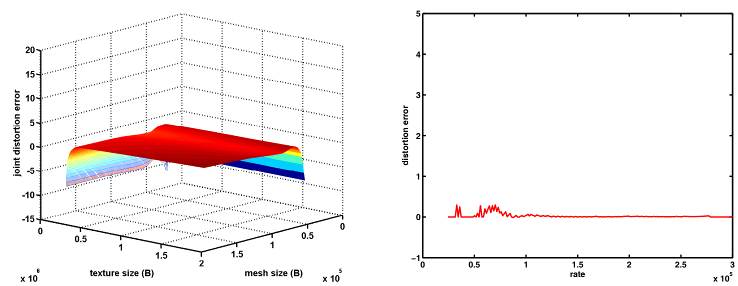
Fig. 3. Comparison of four transmission strategies.



(a) The first layer



(b) The second layer



(c) The third layer

Fig. 4. The approximation error of rate-distortion surfaces and curves.

the joint distortion does not decrease monotonically for the low rate region. Our algorithm can easily solve this problem.

To verify the layered sampling algorithm, we randomly choose a viewpoint and calculate its actual and approximated rate-distortion surfaces. The errors of approximated rate-distortion surfaces and curves are shown in Fig. 4 at various layers of the viewing mesh. The error of the rate-distortion curve indicates how much the approximated rate-distortion surface deviates from the actual one. It is clear that the error decreases when the viewing mesh has more layers.

5. REFERENCES

[1] H. Hoppe, "New quadric metric for simplifying meshes with appearance attributes," in *Visualization '99. Proceedings*, San Francisco, CA, USA, 1999, pp. 59–66, 511.

[2] Michael Garland and Paul S. Heckbert, "Simplifying surfaces with color and texture using quadric error metrics," in *Proceedings of Visualization '98*, 1998, pp. 263–269.

[3] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, pp. 79–93, Jan.–March 2000.

[4] M. D. Adams and F. Kossentini, "JasPer: a software-based JPEG-2000 codec implementation," in *Proceedings of IEEE International Conference on Image Processing*, October 2000, pp. 53–56.