Progressive Transmission of Textured Graphic Model Over IP Networks

1. Research Team

Project Leader:	Prof. C.-C. Jay Kuo, *Electrical Engineering*
Other Faculty:	Prof. Mathieu Desbrun, *Computer Science* Prof. Antonio Ortega, *Electrical Engineering*
Post Doc(s):	Qing Li, Meiyin Shen, Dai Yang, Chia-Hung Yeh,
Graduate Students:	Chao-Hua Lee, Ming-Sui Lee, Jingliang Peng, Sheng Yang
Undergraduate Students:	Hilton Chu
Industrial Partner(s):	Institute of Information Industry, Microsoft

2. Statement of Project Goals

3D graphic transmission has recently attracted great attention in both academic research and commercial applications, such as architectural presentations, virtual marketing, and network gaming, *etc*. In these applications, it is desirable to progressively encode and transmit graphic models and their associated textures in order to reduce the lengthy transmission and rendering time. The textured model transmission consists of three steps: (1) graphic and image compression, (2) multiplexing, and (3) rendering. To reduce the initial waiting time, it is desirable to represent both the mesh and texture data by levels of details. The layered representation enables progressive transmission and rendering. Mesh compression and rendering have been well studied. However, multiplexing is still an open problem and will be studied in this work.

For an intermediate mesh of a compressed graphic model, the distortion comes from three aspects: the geometric error, surface warping, and texture distortion. The geometric error is introduced because of the inaccuracy of vertex positions. Surface warping is a deviation of texture mapping due to the error of texture coordinates. Texture distortion is a distortion of a textured image due to its lossy representation. The purpose of progressive mesh transmission is to maximize the visual quality for every intermediate mesh. Most existing methods [1.2] put the focus on the geometric error. However, the joint optimization of mesh and texture data for progressive transmission is seldom addressed. As a result, the optimality of rendered quality at each stage of progressive transmission cannot be guaranteed. 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 every stage of progressive transmission.

3. Project Role in Support of IMSC Strategic Plan

Techniques of graphic streaming play an important role in IMSC strategic plan, which aims at facilitating the browsing of graphic models via wired or wireless channels. It is desirable to progressively compress and transmit graphic models to reduce the initial wait time. However, it is important to maximize the visual quality with limited network resources. The development and understanding to the quality measurement of levels of details can be used to facilitate controlled quality of service of walkthrough virtual environments. Our work is carried out jointly with quite a few other faculty members in IMSC due to the multi-disciplinary nature of the proposed research.

4. Discussion of Methodology Used

An optimized scheme of multiplexing coded mesh and texture data to facilitate progressive transmission of 3D textured models is proposed in this work. The mesh and texture data of a 3D textured model are fed into their respective compression modules and represented by a series of levels of details. Then, for a given viewpoint, a rate-distortion surface can be generated based on the multiplexing of mesh and texture data in different details. The distortion is calculated by measuring the visual quality of rendered images. Furthermore, an optimal path over the rate-distortion surface is determined by the steepest descent algorithm. When the mesh and texture data streams are transmitted in a certain ratio along the optimal path, it is guaranteed that the rendered images have the best visual quality perceived from the given viewpoint. To deal with an arbitrary viewpoint, we propose a layered sampling algorithm so that the optimal path can be interpolated from finite sampling points. Experimental results demonstrate that the proposed method can provide the best visual quality of a textured 3D model for any observation point at any bit rate.

5. Short Description of Achievements in Previous Years

We studied progressive transmission technique of textured graphic models. This is a challenging problem, since there is no simple metric to measure the distortion of compressed textured graphic models. Therefore, we proposed a novel algorithm based on rate-distortion surface to solve this problem.

5a. Detail of Accomplishments during the Past Year

<u>Problem Definition</u>

Although the mesh and texture data can be easily progressively compressed by the respective codecs, the multiplexing of mesh and texture data into one bit stream has not yet been addressed before. This issue is critical to progressive textured model delivery.

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

$D_m = f_m(R_m)$,

$D_t = f_t(R_t)$,

where D_m and D_t are the mesh and texture distortion functions, and R_m and R_t are the mesh and the texture bit rates, respectively.

Let us take the 'Bunny' model as an example, its rate-distortion curves of mesh and texture are illustrated in Figure 1. Typically, the mesh distortion is measured in terms of the displacement of 3D mesh vertex points while the texture distortion is measured in the difference of the gray-level values defined on a 2D grid. When we want to multiplex these two progressive bit streams into one, it is a nontrivial problem to define a distortion measure for the textured graphic model.

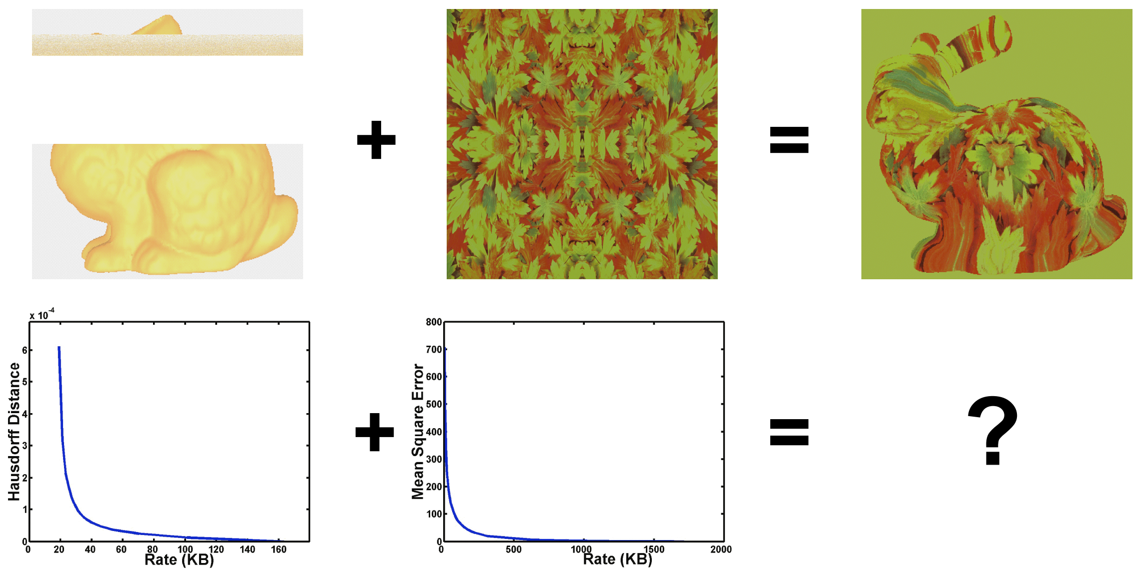


Figure 1 The difficulty of defining a distortion metric for a textured graphic model.

Rate-Distortion Surface

To solve the mesh-texture multiplexing problem, we propose a new concept called rate-distortion surface to describe how the display quality of textured models varies with rate. Without loss of generality, we compress the mesh data using the edge collapse/vertex split method [8.9] and the texture data using the JPEG2000 image compression standard [10] in this research. However, it is worthwhile to point out that the framework adopted in this study is generally applicable to any progressive mesh and texture codecs. The adopted specific codecs serve as a concrete example for the ease of discussion.

It is assumed that the mesh and texture data are, respectively, represented by m and n layers as

$$M_0 \subset M_1 \subset \cdots \subset M_{m-1},$$

$$T_0 \subset T_1 \subset \cdots \subset T_{n-1}.$$

Thus, there are $m \times n$ combinations of different mesh and texture resolutions:

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

For the distortion measure, we first consider the case where a viewer examines the model from a fixed viewpoint. Then, to measure the joint distortion associated with M_iT_j with $0 \le i < m$ and $0 \le j < n$, 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.* $M_{m-1}\ T_{n-1}$. For any combination, we can obtain such a distortion measure. Finally, the distortion values for all possible combinations form a surface, called rate-distortion surface, which is illustrated in Figure 2.

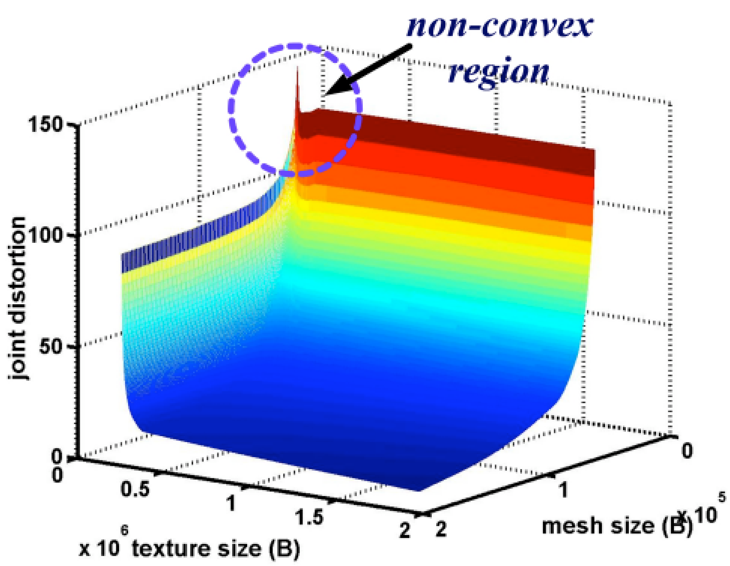


Figure 2 The rate-distortion surface.

We implemented four distortion metrics in the experiments; namely, MSE, TM, Mannos-Sakrison's filter, and Daly's filter. The former two are the spatial domain metrics and the latter two are the spatial-frequency domain metrics. We only present the results of Mannos-Sakrison's filter [11] and MSE since the two in the pair have similar performance. The rate-distortion surface can be calculated off-line to save viewer's waiting time. It is worthwhile to point out that illumination could change the distortion function also. However, since it is generated on rendered images, it is impossible to pre-calculate the rate-distortion surface with respect to the illumination effect.

Based on Phong's lighting model, the luminance of one given point P on the model surface can be expressed by

$$I = k_a i_a + \sum_{i=1}^{M} f_i i_{li} \left[k_d \left(\vec{N} \cdot \vec{L_i} \right) + k_s \left(\vec{N} \cdot \vec{H_i} \right) \right], \tag{1}$$

where k_a, k_d, and k_s are the coefficients of the ambient, diffuse, and specular lights. i_a is the intensity of the ambient light. i_{di} and i_{si} are the intensities of the diffuse and specular components of the ith light source. f_i is the attenuation coefficient of the ith light source. $\overrightarrow{N}$ is the normal vector at P. $\overrightarrow{L_i}$ is the vector pointing from P to the ith light source. We have

$$\overrightarrow{H_i} = \frac{\overrightarrow{L_i} + \overrightarrow{V}}{2}$$

where $\overrightarrow{V}$ is the viewing vector. For clarity, we denote the three parts of the above equation as

$$I_a = k_a i_a,$$

$$I_{di} = f_i i_{li} k_d \left(\overrightarrow{N} \cdot \overrightarrow{L_i}\right),$$

$$I_{si} = f_i i_{li} k_s \left(\overrightarrow{N} \cdot \overrightarrow{H_i}\right)^n.$$

Since k_a is a constant, the luminance is proportional to the intensity of the ambient light. Therefore, the influence of the ambient light on the rate-distortion surface can be easily obtained. For diffuse and specular components, the calculation is much more complicated. We discuss them with two cases below.

First, only the intensities of light sources vary while the viewing positions remain unchanged. Since $\overrightarrow{N} \cdot \overrightarrow{L_i}$ remains invariant, the diffuse component of the rate-distortion surface only depends on the intensity of the diffuse light and can be solved in a similar way to the ambient light. Furthermore, considering that the calculation of a rate-distortion surface implies a fixed viewpoint, we can claim that $\overrightarrow{N} \cdot \overrightarrow{H_i}$ is a constant as well. Therefore, the specular component can be derived similarly. The rate-distortion surface under fixed illumination is calculated via

$$D_j^{(0)} = D_a^{(0)} + D_d^{(0)} + D_s^{(0)},$$

where $D_a^{(0)}$, $D_d^{(0)}$ and $D_s^{(0)}$ denote the ambient, diffuse, and specular components, respectively. When the intensities of light sources vary, the rate-distortion surface can be calculated via

$$D_j^{(1)} = c_a D_a^{(0)} + c_d D_d^{(0)} + c_s D_s^{(0)},$$

where $D_j^{(1)}$ is the joint rate-distortion surface under the new illumination scenario. The coefficients are calculated via

$$c_a = \frac{i_a^{(1)}}{i_a^{(0)}},$$

$$c_d = \frac{\sum_{i=1}^{M} f_i^{(1)} i_{li}^{(1)}}{\sum_{i=1}^{M} f_i^{(0)} i_{li}^{(0)}},$$

$$c_s = c_d.$$

For the second case, both the intensities and positions vary. It is difficult to directly derive the rate-distortion surface under any illumination in this case. We leave this to our future work. In this work, we only pay attention to the first case.

Optimal Path

In practice, we have the integer constraint on the mesh layer index i and the texture layer index j, as shown in Figure 3. 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 are completely delivered to reach the full resolution. The result is accurate if the gird size is small enough. The mesh and texture gradient surfaces are illustrated in Figure 4 (a) and (b).

To demonstrate the performance of the proposed multiplexing scheme, we compare it with the following three strategies.

- 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.
- The mesh-first scheme: The mesh data are transmitted first with the minimum amount of the texture data. Then, more texture refinement data are transmitted after all mesh data have been completely delivered.
- The texture-first scheme: The texture data are transmitted first with the minimum amount of the mesh data. Then, more mesh refinement data are transmitted after all texture data have been completely delivered.

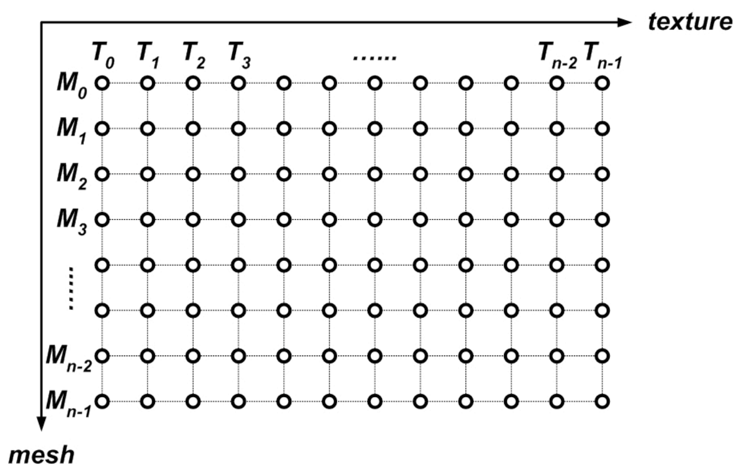
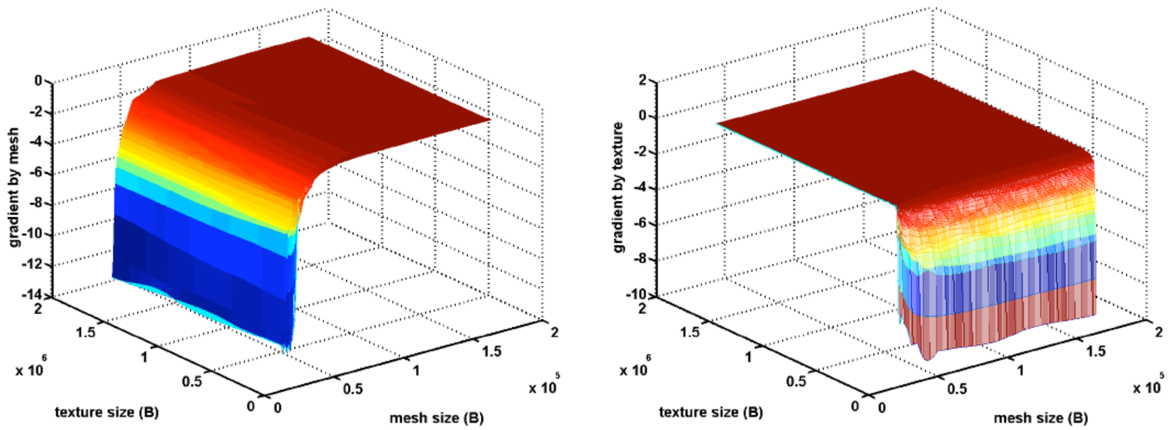


Figure 3 The mesh texture grid.

The proposed multiplexing 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 sometime 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 a 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.



(a) Gradient due to mesh change (b)Gradient due to texture change

Figure 4 The gradient surfaces.

The curve shown in Figure 5 describes the amount of mesh data that should occupy among the total bit rate. This curve guides how to allocate bits for mesh and texture data representation during progressive transmission to achieve the best display quality with respect to a given viewpoint. To reduce the cost of storing the data for this curve, we may approximate the curve by a low order polynomial. In particular, we pay special attention to the interval where both the mesh and the texture data have not been completely delivered. It is observed that the curve can be well approximated by a third-order polynomial function. Please note that the performance of the approximation can be further improved by adopting weighted curve fitting, since the data have larger influence on visual quality in the beginning. The weights can be determined according to the rate-distortion curve, which can be a good topic for future research.

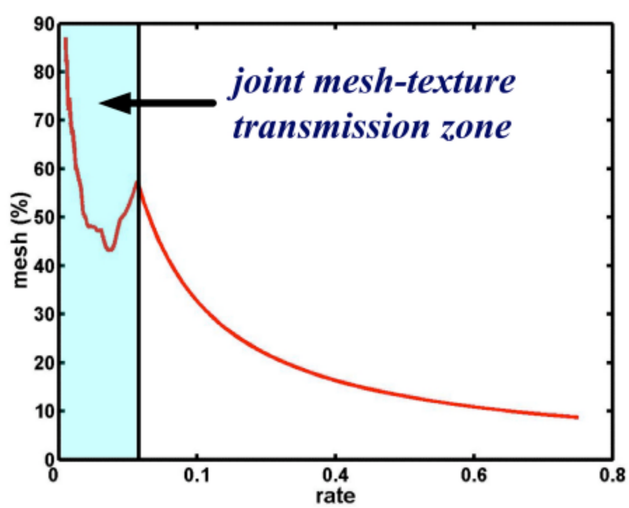


Figure 5 The mesh percentage curve.

Continuity of Distortion Surface

Before presenting the sampling algorithm, we first prove that the image-based distortion function is continuous with respect to viewer's motion. Based on this fact, we can argue that, if the sampling is fine enough, the rate-distortion surface for an arbitrary viewpoint can be reconstructed from its nearby sampling points.

It is assumed that the textured graphic model is captured by a pinhole camera C that has a finite angular resolution as illustrated in Figure 6. Then, the pixel value of the capturing camera is an integral of the illumination of the light arriving at the camera plane. The surface of the given model is represented by

$$m = f_{mesh}(s), \quad a \le s \le b,$$

where a and b are the parametric coordinates of A and B, respectively. The sectors P_0CP_1 and $P_0'C'P_1'$ denote the fields of view of the camera before and after the change of the viewing parameters. Therefore, the illumination of one point s in the surface can be calculated by (1).

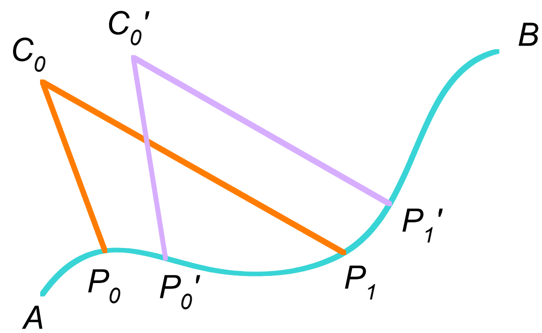


Figure 6 The pixel value calculation.

The pixel value before and after camera movement is calculated by

$$P(s_0, s_1) = \int_{s_0}^{s_1} I(s)\,ds, \qquad (2)$$

$$P(s_0', s_1') = \int_{s_0'}^{s_1'} I(s)\,ds. \qquad (3)$$

Thus, the pixel value difference can be calculated by subtracting (3) from (2)

$$\begin{aligned} p_d &= P(s_0', s_1') - P(s_0, s_1) \\ &= \left(\int_{s_1}^{s_1'} - \int_{s_0}^{s_0'} \right) I_a(s)\,ds + \sum_{i=1}^{M} \left(\int_{s_1}^{s_1'} - \int_{s_0}^{s_0'} \right) I_{di}(s)\,ds \\ &\quad + \sum_{i=1}^{M} \left(\int_{s_1}^{s_1'} - \int_{s_0}^{s_0'} \right) I_{si}(s)\,ds \end{aligned}$$

Since $0 \le k_a(s) \le 1$, we have $0 \le k_a(s) i_a \le i_a$. Then, we can obtain

$$\left| \int_{s_1}^{s_1'} I_a(s)\,ds \right| \le i_a |s_1' - s_1|, \quad \left| \int_{s_0}^{s_0'} I_a(s)\,ds \right| \le i_a |s_0' - s_0|,$$

and

$$\left| \left(\int_{s_1}^{s_1'} - \int_{s_0}^{s_0'} \right) I_a(s)\,ds \right| \le i_a \left(|s_1' - s_1| + |s_1' - s_1| \right).$$

Let $\Delta_1 = |s_1' - s_1|$ and $\Delta_0 = |s_0' - s_0|$. Then, the above inequality can be written as

$$\left|\left(\int_{s_1}^{s_1'} - \int_{s_0}^{s_0'}\right) I_a(s)ds\right| \le i_a(\Delta_0 + \Delta_1)$$

Similarly, since $0 \le k_d(s), k_s(s) \le 1$, $\left|\vec{N}(s)\cdot\vec{L}_i(s)\right| \le \left|\vec{N}(s)\right|\cdot\left|\vec{L}_i(s)\right| = 1$ and $\left|\vec{N}(s)\cdot\vec{H}_i(s)\right| \le \left|\vec{N}(s)\right|\cdot\left|\vec{H}_i(s)\right| = 1$, we obtain

$$\left|\left(\int_{s_1}^{s_1'} - \int_{s_0}^{s_0'}\right) I_{di}(s)ds\right| \le f_i i_{li}(\Delta_0 + \Delta_1),$$

$$\left|\left(\int_{s_1}^{s_1'} - \int_{s_0}^{s_0'}\right) I_{si}(s)ds\right| \le f_i i_{li}(\Delta_0 + \Delta_1).$$

Finally, we have the following inequality:

$$\left|P(s_0', s_1') - P(s_0, s_1)\right| \le \left(i_a + 2\sum_{i=1}^{M} f_i i_{li}\right)(\Delta_0 + \Delta_1) \qquad (4)$$

The above inequality says that the pixel value difference is bounded when the viewer moves along at a small distance. Accordingly, the distortion function is continuous with viewer's motion. Thus, the rate-distortion surface can be reconstructed within a certain error bound from its nearby sampling points.

Sampling Function

With the method described above, we successfully solve the problem of mesh-texture multiplexing when the viewer looks at the model from a fixed viewpoint. However, it is typical that the viewer will move around and observe the model from different viewpoints. We propose an adaptive layered sampling algorithm to address this problem.

Here, we only present the work when the viewing distance between the viewer and the model remains unchanged. The solution can be generalized to the varying distance case.

As a result of the constant viewing distance, the set of all viewpoints forms a sphere, which is called *viewing sphere*. The idea of layered sampling is to approximate the viewing sphere by a progressive triangle mesh, called *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 to subdivide the viewing mesh until it is fine enough. The goal of viewing mesh construction is to iteratively insert new viewpoints so that the rate-distortion surface of any 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, their rate-distortion surface can be denoted by

$$D_{j0} = f_{j0}(R_m, R_t),$$
$$D_{j1} = f_{j1}(R_m, R_t),$$
$$D_{j2} = f_{j2}(R_m, R_t)$$

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

$$D_j' = \left(\|vp - vp_0\| \cdot D_{j0} + \|vp - vp_1\| \cdot D_{j1} + \|vp - vp_2\| \cdot D_{j2}\right) / \left(\|vp - vp_0\| + \|vp - vp_1\| + \|vp - vp_2\|\right),$$

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

$$e = \left\{ \iint \left[D_j(R_m, R_t) - D_j'(R_m, R_t) \right] dR_m dR_t \right\} / (R_M R_T), \tag{5}$$

where R_M and R_T denote the total size of mesh and texture data, respectively.

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

To reduce the approximation error as fast as possible, the viewing mesh construction process should meet the following two requirements. First, triangles in each layer should be approximately of the same size and a similar shape. Second, the sides of the triangle should be approximately of the same length. In the proposed viewing mesh subdivision method, we first find the midpoint of each edge and extend it to intersect with the surface of the viewing sphere. The points of intersection are viewed as new viewpoints. Then, we subdivide one triangle into four by connecting the newly inserted viewpoints. The viewing mesh subdivision process is shown in Figure 7.

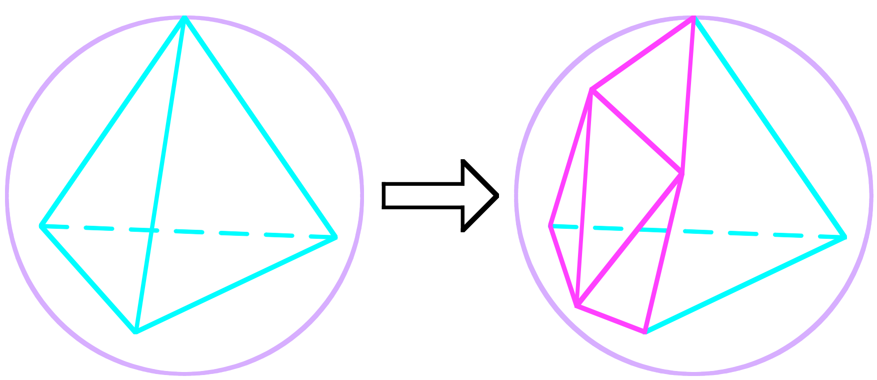


Figure 7 The viewing mesh subdivision process.

With the above adaptive layered sampling algorithm, we obtain the rate-distortion surfaces at sampling points. By expressing the sampling points with 3D angular coordinates, we have the following:

$$D_{\text{int}} = g(\square, \square),$$

where D_{int} is the integral over a rate-distortion surface in viewpoint (□,□). An example of the sampling surface is illustrated in Figure 8. By using the sampling surface, the rate-distortion function of an arbitrary viewpoint can be calculated.

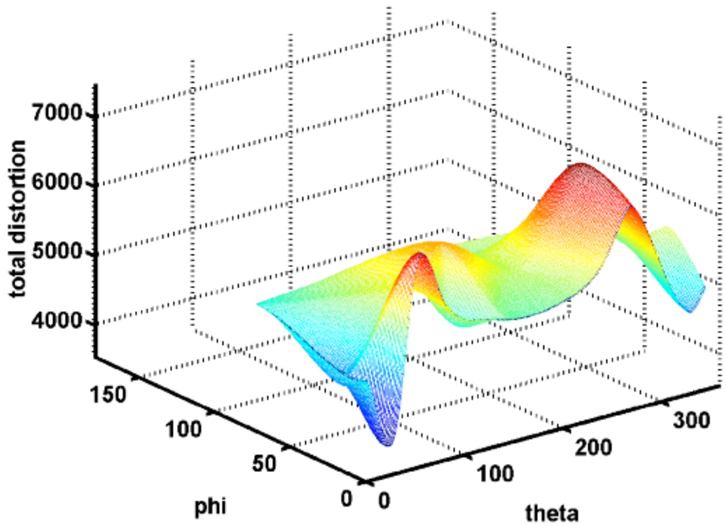


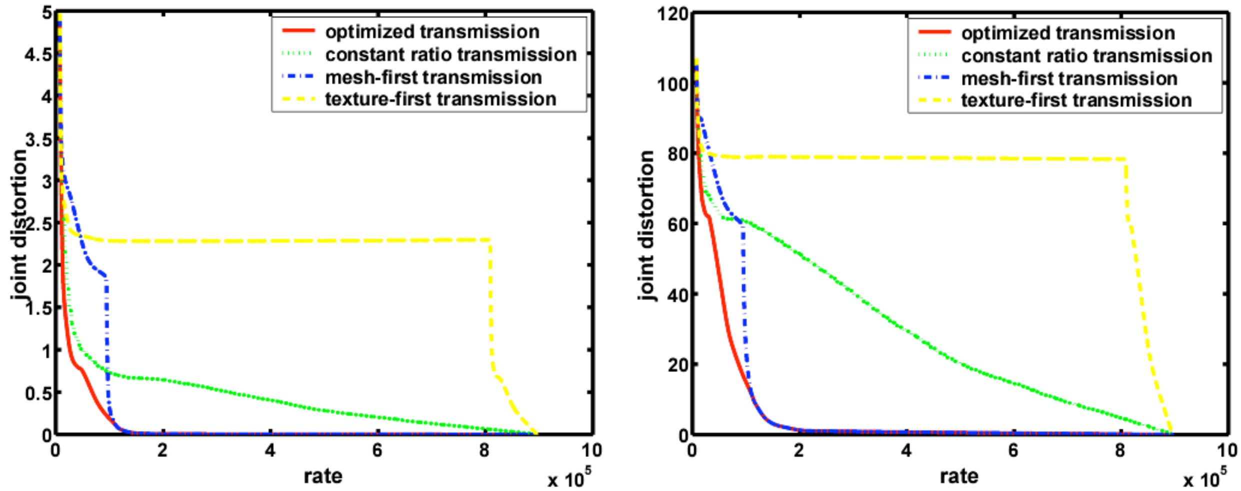
Figure 8 The sampling surface.

Experimental Results

In this section, we test the proposed algorithm in two stages. First, we test if the rate-distortion surface can correctly reflect the relationship between the bit rate and the display quality of textured models to offer the best multiplexing strategy. Second, we test if the sampling function works well for the case when the viewer looks at the model from an arbitrary viewpoint. As mentioned before, to test the performance of the proposed algorithm, we compare it with other three strategies: the constant ratio scheme, the mesh-first scheme, and the texture-first scheme.

The rate-distortion curves of 'Angelfish' by Mannos-Sakrison's filter and MSE are illustrated in Figure 9. We see from the figures that the proposed scheme always outperforms the other three schemes, no matter what distortion metrics are adopted. As mentioned above, when the mesh resolution is very low, the increasing resolution of texture may sometimes increase the joint distortion. This is observed in the texture-first curve. Therefore, when the textured model is transmitted by the constant ratio or the texture-first schemes, the joint distortion may not decrease monotonically. The image rendered in the full resolution of both mesh and texture is shown in Figure 10.

When the angelfish model is transmitted with 10% of the total number of bits for the full resolution, the rendered images and residual images under four multiplexing methods are shown in Figures 11 and 12, respectively. For the proposed method, the optimal path selection is based on the distortion measured with Mannos-Sakrison's filter. We observe that the poor quality of the mesh-first method mainly lies in the texture component, while that of the texture-first method mostly exists in the silhouette of the object, *e.g.* the tail part. The distortion of images using the optimized multiplexing transmission is hardly visible. We can draw a similar conclusion for the bunny model. Although the images of the mesh-first method look apparently worse than those of the texture-first method, their MSE distortion values sometimes (for example, 'Angelfish' and 'wolf' model) are smaller. On the other hand, the results of Mannos-Sakrison's filter are more consistent with human perception in most cases. This is why people feel that the spatial-frequency domain metrics perform better than spatial domain metrics.



(a) Mannos-Sakrison's filter (b) MSE

Figure 9 Rate-distortion curves.

Table I. Distortions of images obtained with 10\% of total bits.

Model	Strategy	M-S's filter	MSE
Angelfish	Optimal	0.1244	16.285
	Constant Ratio	0.7653	61.855
	Mesh-First	1.6063	59.505
	Texture-First	1.2689	78.662
Bunny	Optimal	0.4536	91.820
	Constant Ratio	1.5588	173.482
	Mesh-First	8.6076	372.848
	Texture-First	6.7602	304.323
Wolf	Optimal	0.0271	10.9108
	Constant Ratio	0.2566	24.3743
	Mesh-First	6.3134	46.5653
	Texture-First	5.8419	229.3847

To verify the adaptive layered sampling algorithm, we randomly choose a viewpoint and calculate its actual and predicted rate-distortion surfaces in different layers of viewing mesh. Based on the two surfaces, we determine two bit allocation results. The prediction error of rate-distortion surfaces and curves are illustrated in Figure 13. The errors shown in the figures are calculated by (5). The error can be easily controlled by viewing mesh subdivision according to user's requirements.

Figure 10. The Angelfish rendered in full resolution.

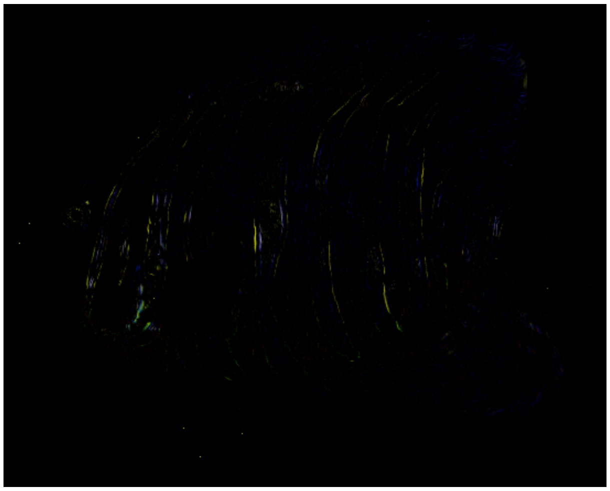
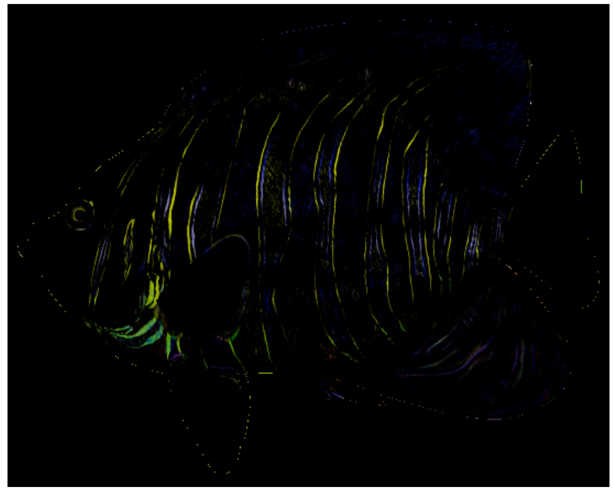
(a) The proposed method

(b) The constant ratio method

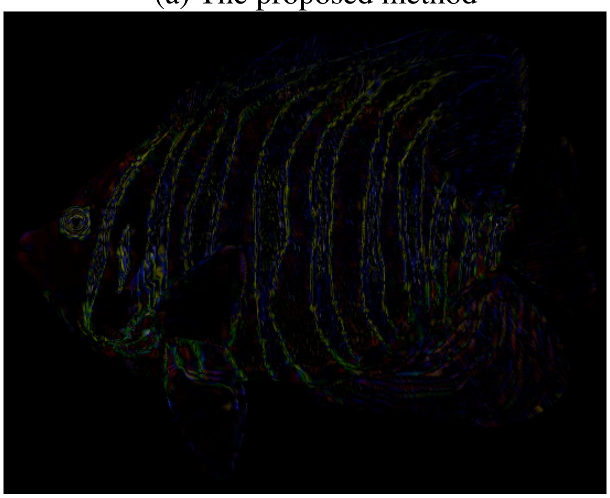
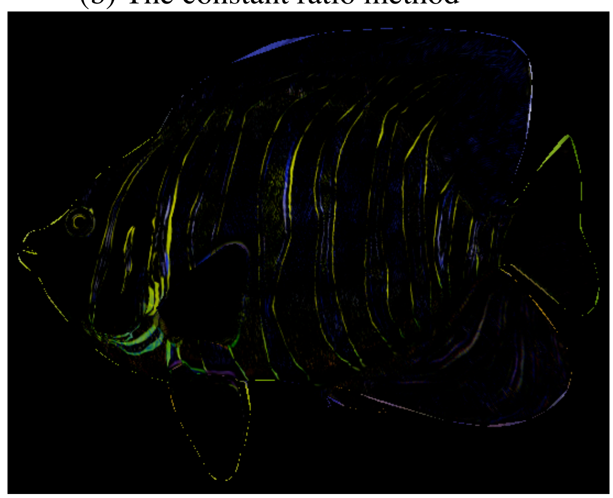
(c) The mesh-first method

(d) The texture-first method

Figure 11 Rendered images of the angelfish model with 10% of the total no. of bits.

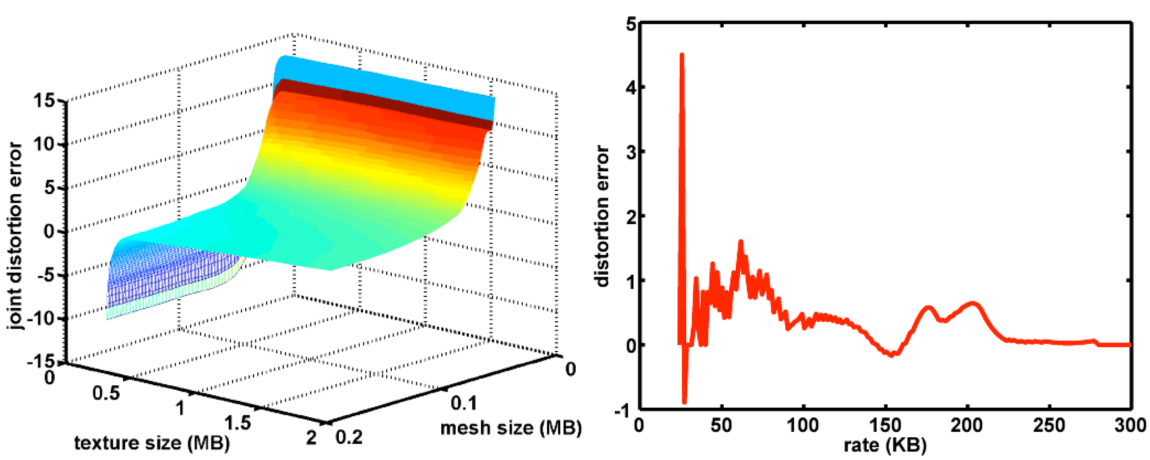



(a) The proposed method (b) The constant ratio method

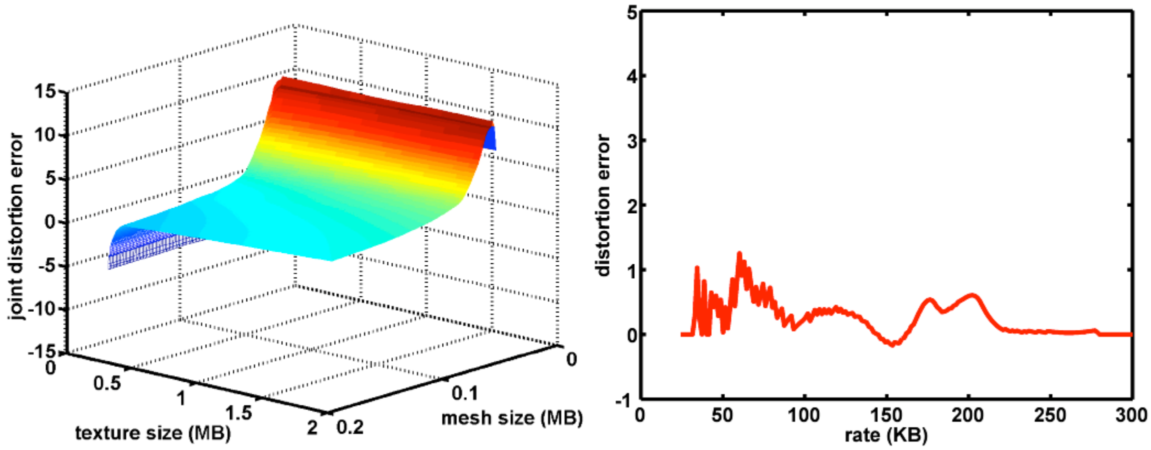



(c) The mesh-first method (d) The texture-first method

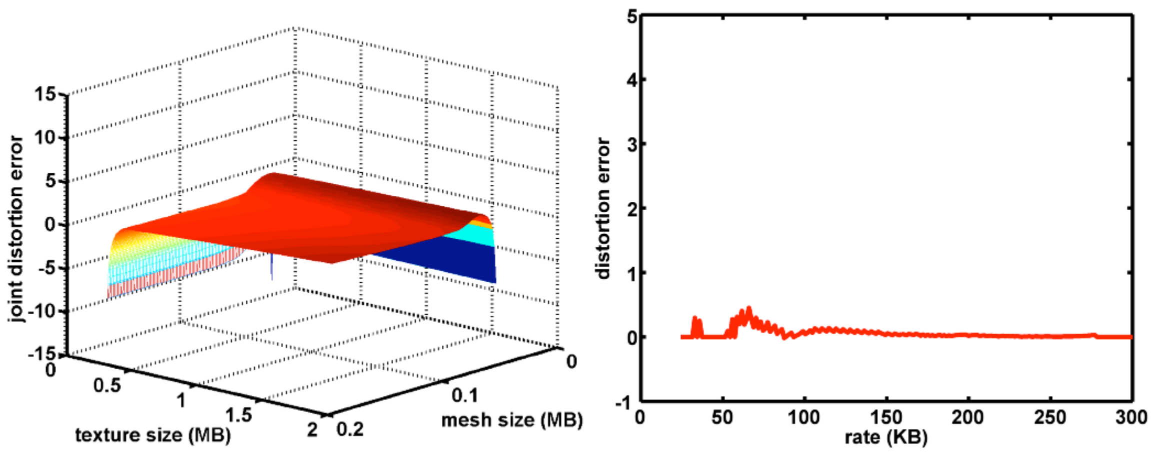
Figure 12 Residual images of the angelfish model.



(a) e=13.2051



(b) e=10.2283



(c) e=2.2323

Figure 13 Rate-distortion curves.

6. Other Relevant Work Being Conducted and How this Project is Different

Most previous work on progressive graphic coding dealt with the mesh data alone, where only the geometric error and surface warping are relevant [1-5]. All the above work considered progressive coding of the mesh data under the assumption that the texture is represented by a full resolution image. Some work paid attention to progressive coding of both geometry and texture data, which is closer to our objective, [6.7]. However, the metric used was usually derived from user's preference about the importance of the geometry and texture information. Thus, the user-dependent approach does not provide an automatic solution and may not be convenient in many applications. Furthermore, it fails to achieve an optimal performance, if there exists a proper metric for performance measure.

To solve the problem, we propose and implement a scheme for progressive transmission of textured graphic models. The main contributions of this work include the following.

- A new concept called rate-distortion surface is proposed to describe the relationship between the bit rate and the display quality of a textured graphic model at a given viewpoint.

- An optimal strategy is developed to multiplex the mesh and the texture bit streams into one single stream for progressive transmission to present the best display quality subject to the bandwidth constraint at a given viewpoint.
- A layered sampling method is proposed so that the multiplexing scheme can be generalized to deal with an arbitrary viewpoint.

7. Plan for the Next Year

Our plan for the next year includes the following:

- Develop progressive rendering algorithm, *i.e.* only update the regions with newly incoming mesh or texture data, for some client devices with limited computational capability.
- Develop more robust strategy to determine the optimal path over rate-distortion surface.

8. Expected Milestones and Deliverables

An optimal mesh-texture multiplexing codec has been implemented by C++. It has the following functions: (1) progressive mesh compression; (2) progressive texture compression; (3) rate-distortion surface calculation; (4) optimal path determination; and (5) adaptive layered sampling. Several conference papers have been published on this topic. There are several long-term challenging issues worth our further study. They include: progressive rendering, optimal path determination, etc. We will continue to investigate these problems in next year and seek some major breakthrough in engineering science. The milestone chart is given below.

9. Member Company Benefits

Both III (Institute of Information Industry) and Microsoft Research Lab show a strong interested in the developed technologies. Demos have been made to them.

10. 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] Jonathan Cohen, Marc Olano, and Dinesh Manocha, “Appearance-preserving simplification,” in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 115–122.

[4] Peter Lindstrom and Greg Turk, “Image-driven simplification,” *ACM Transactions on Graphics*, vol. 19, pp. 204–241, July 2000.

[5] Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe, “Texture mapping progressive meshes,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, August 2001, pp. 409–416.

[6] M. Okuda and T. Chen, “Joint geometry/texture progressive coding of 3d models,” in *Proceedings of Int’l Conf. on Image Processing*, 2000, pp. 632–635.

[7] María José Abásolo and Francisco Perales, "Geometric-textured bitree: Transmission of a multiresolution terrain across the internet," *Journal of Computer Science & Technology*, vol. 2, no. 7, pp. 34–41, Oct. 2002.
[8] H. Hoppe, "Progressive meshes," in *Computer Graphics Proceedings*, New Orleans, August 1996, Annual Conference Series, pp. 99–108.
[9] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, pp. 79–93, Jan.–March 2000.
[10] 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.
[11] J. L. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Transactions on Information Theory*, vol. 10, no. 5, pp. 525–536, 1974.