FQM: A Fast Quality Measure for Efficient Transmission of Textured 3D Models

Dihong Tian
School of Electrical and Computer Engineering
Georgia Tech Regional Engineering Program
Georgia Institute of Technology
dhtian@ece.gatech.edu

Ghassan AlRegib
School of Electrical and Computer Engineering
Georgia Tech Regional Engineering Program
Georgia Institute of Technology
gregib@ece.gatech.edu

ABSTRACT

In this paper, we propose an efficient transmission method to stream textured 3D models. We develop a bit-allocation algorithm that distributes the bit budget between the geometry and the mapped texture to maximize the quality of the model displayed on the client's screen. Both the geometry and the texture are progressively and independently compressed. The resolutions for the geometry and the texture are selected to maximize the quality for a given bitrate. We further propose a novel and fast quality measure (FQM) to quantify the perceptual fidelity of the simplified model. Experimental results demonstrate the effectiveness of the proposed bit-allocation algorithm using FQM. For example, when the bit budget is 10KB, the quality of the ZEBRA model is improved by 15% using the proposed method compared to distributing the bit budget equally between the geometry and the texture.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three dimensional graphics and realism; I.4.7 [Image Processing and Computer Vision]: Feature Measurement; E.4 [Data]: Coding and information theory; C.2.m [Computer-Communication Networks]: Miscellaneous

General Terms: Algorithms, Measurement

Keywords: 3D model, Texture, Geometry, Bit-allocation, Quality measure

1. INTRODUCTION

A well known technique in 3D graphics is texture mapping where an image is mapped to a mesh surface. Texture mapping is effective when desired surface details are expensive to achieve by solely using geometry. The nature of texture mapping and the projection of the texture on the screen complicate the compression of textured models. Essentially, the geometric accuracy of the mesh may no longer be the primary fidelity metric since they can be compensated by the mapped texture. The problem becomes more complex when the textured model is transmitted in a resource constrained environment such as bandwidth-limited channels or using handheld devices with limited rendering capability. In these cases, the textured model needs to be compressed into a hierarchical bitstream where the number of transmitted bits to the client depends on the available resources.

The two components of a textured model, i.e., the geometry and the mapped texture, differ in nature and therefore the compression algorithms to be applied to these two components are different. In this paper, we are not proposing new compression algorithms but instead we study how to distribute the bit budget between the geometry and the mapped texture. We limit this work to 2D mapped textures such as images even though the framework can be extended to other textures such as video sequences. The problem this paper addresses can be stated as follows: *Given a textured 3D model to be transmitted over a bandwidth-limited channel, how to distribute the bit budget between the geometry and the mapped texture in order to maximize the quality of the model that is displayed on the client's screen.*

We first conduct progressive compression for the geometry using the *texture deviation* metric to preserve appearance [4] and a wavelet based coder for the mapped texture [11]. Then we find the proper LODs for the geometry and the texture, respectively, such that the best perceptual fidelity is achieved while satisfying the bit constraint. Apparently, the core of the bit-allocation algorithm is how to properly predict the visual fidelity when substituting simplified textures to coarse mesh geometry. In this paper, we propose a novel quality measure to quantify the error incurred during model simplification. Quality degradation that is incurred by geometric error and texture distortion is jointly modelled through an *equalization* factor. Depending on the features of the geometry and the mapped texture, the equalization factor is estimated as a constant using error samples conducted in the screen space. The proposed quality measure is computationally efficient, and the experiments verify its effectiveness in successfully quantifying the visual fidelity.

The paper is organized as follows. Section 2 summarizes the relevant prior art. Section 3 describes the progressive compression methods. The process of measuring the quality of simplified models and the proposed fast quality measure (FQM) are explained in Section 4. The optimal bit allocation framework and the heuristical algorithms are given in Section 5. We discuss the experimental results in Section 6 and the conclusion remarks are summarized in Section 7.

MM'04, October 10-16, 2004, New York, New York, USA.

2. RELATED WORK

The challenge in simplifying a 3D surface is to preserve the appearance attributes while generating the lower-polygon-count mesh [6]. The traditional simplification criterion, which measures the approximation accuracy using the surface distance [10, 5], may not produce satisfying approximation for textured models because the geometric error can either be concealed or be highlighted after texture mapping. Cohen et al. introduced in [4] *texture deviation* as an appropriate criterion to measure the surface distortion resulting from simplifying texture mapped meshes. The texture deviation incorporates the texture domain and the geometry through the parametric correspondence, imposing a stricter error metric on mesh simplification. This error metric is employed in our mesh encoder, details presented in Section 3.

Measuring visual fidelity has become crucial in the fields of model simplification and level-of-detail (LOD) control. A study of techniques for measuring and predicting visual fidelity was conducted by Watson et al. in [14], where they examined experimental (subjective) techniques as well as several automatic (computational) techniques including Metro tool [3] and mean squared image error (MSE). Both Metro and MSE were evaluated successful predictors of quality as judged by human ratings. Nonetheless, the experimental study in [14] was performed on models with geometry only.

Joint consideration of the geometry and the texture is necessary when coding a textured model and under a resource constrained environment [1, 9]. To the best of our knowledge, the most closely related effort that addressed this challenge is that of Balmelli [2], where he studied joint mesh and texture compression for terrain models. He simplified both the geometry and the mapped texture of the terrain model and then measured the quality of the textured models at various LODs using the screen-space error, i.e., the peak signal-to-noise ratio (PSNR) of the rendered image. Even though this is a pioneering work, it is limited to meshes with subdivision connectivity such as terrain models. In contrast, in this paper, we generalize the framework to meshes with any connectivity and furthermore, we propose a novel visual fidelity measure that is drastically faster than the traditional screen-space error (PSNR). In addition, the proposed quality measure does not depend on the viewpoint. Before we explain the quality measure, we first present an overview of the proposed compression system for the textured model in the following section.

3. SYSTEM OVERVIEW

The block diagram of the proposed system is illustrated in Figure 1. The 3D mesh is progressively compressed to generate a base mesh and a number of enhancement layers. The client download the base mesh first and then the enhancement layers are downloaded to refine the base mesh. Similarly, the texture image is compressed into a scalable bitstream. The quality of the decoded texture at the client side increases as more bits are downloaded from the texture bitstream. Driven by a fidelity measure, the"bit allocation" block optimally assigns coding bits between the mesh and the texture. The output is transmitted over a bandwidth limited channel, aiming to provide the best rendering quality on the client side while satisfying the bitrate constraint.

There are several methods available for coding textures and we do not plan to devote effort toward developing new ones. Instead, throughout the paper we limit the texture to be typical 2D images, and work with published image compression algorithms. More specifically, we adopt the wavelet-based compression algorithm known as SPIHT (Set Partitioning in Hierarchical Trees) [11] to encode the texture. In the remainder of this section, we explain in detail the process of coding the parameterized mesh.

The multi-resolution mesh encoder consists of several components. The vertex and texture coordinates are first quantized to facilitate the successive processes. In surface approximation, we progressively simplify the mesh by applying consecutive edge-collapse operations [6]. As a consequence of edge-collapses, the texture coordinates need to be updated accordingly. For simplicity, in our scheme we perform the *half-edge* collapse operation, which affects only the neighborhood of one vertex of the to-be-collapsed edge and leaves the position and attribute of the other vertex unchanged. Most other placement variants, such as the midpoint placement of the split-vertex, do not guarantee that the coordinates of the split-vertex stay on the quantized coordinate grid and require computation of new texture coordinates, making the encoding process more complex.

For better preservation of the appearance, the edge-collapse operations are performed in the order of increasing error according to the *texture deviation* metric [4]. The texture deviation measures the cost of an edge collapse as the maximum distance in the geometric space from points on the simplified mesh to their correspondents on the input surface. As illustrated in Figure 2, V_{i-1} and V_i are 3D points on the two meshes before and after an edge collapse, respectively, both having the same texture coordinates, v; the incremental texture deviation of this edge-collapse is defined as

$$\max_{v \in P} E_{i,i-1}(v) = \max_{v \in P} \parallel F_i^{-1}(v) - F_{i-1}^{-1}(v) \parallel,$$

where F_i and F_{i-1} are texture mapping functions.

For fast computing, we exploit a heuristic that measures the incremental texture deviation rather than using the conservative bounds of accumulated error as in [4]. For a half-edge collapse operation, $(V_1, V_2) \rightarrow V_1$, we expect the maximum deviation to lie either at the removed vertex V_2 or at an edge-edge intersection point in the parametric neighborhood (the yellow points in Figure 2, for example). This heuristic is akin to "memoryless" error that has proven effective for 3D geometry compression in [8], and has been empirically confirmed working well in [12].

The base mesh resulting from progressive simplification is compressed using the surgery topological algorithm [13] and the refinement information is encoded similarly following the *compressed progressive mesh* (CPM) algorithm [10]. We treat each vertex as a vector $V \in \mathbf{R}^5$. The first three components of V consist of spatial coordinates, and the remaining two components are texture coordinates. We then compress the edge-collapses using vertex prediction followed by entropy coding. In the rest of the paper, we refer to the above scheme of progressive mesh coding as texture-deviation based CPM algorithm, or simply, TD-CPM.

4. PERCEPTUAL FIDELITY MEASURE

Using the TD-CPM algorithm, we generate a family of multi-resolution meshes, $\{M_i\}_{i=0\ldots n}$, with resolution decreasing from the original surface M_0 to the base mesh M_n. Likewise, using progressive coding we produce a family of texture images with varying resolutions denoted by $\{T_j\}_{j=0\ldots m}$.

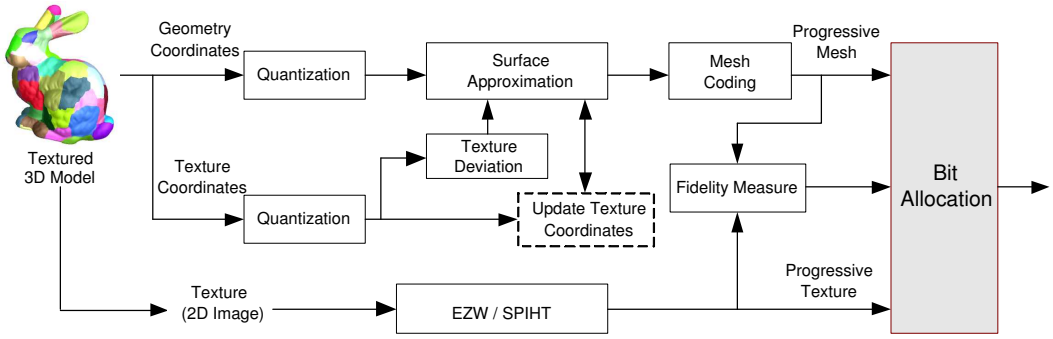


Figure 1: A block diagram of the encoder.

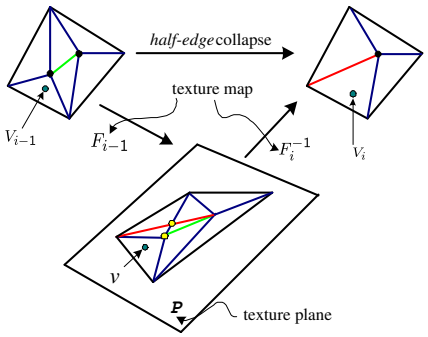


Figure 2: Illustration of the texture deviation criterion.

Each pair of mesh and texture resolutions, (M_i, T_j), will give an approximated representation of the original textured 3D model. Each pair, (M_i, T_j), differs from the original textured 3D model by certain error. It is crucial to measure such error so as to properly reflect the perceptual fidelity. In this section, we discuss the methods of quantifying such difference. We first discuss the traditional method of evaluating the perceptual fidelity using the image-space error. Then, we propose a *fast quality measure* (FQM) that provides meaningful evaluation of visual quality for different pairs of mesh and texture resolutions and works drastically faster than the traditional screen-space error measure.

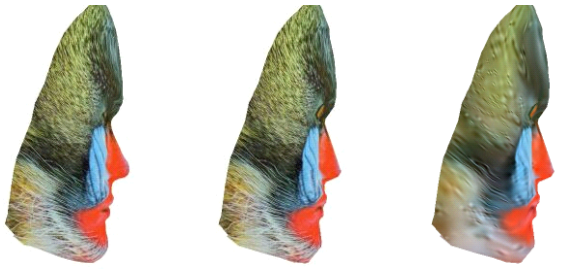
4.1 The Screen-Space Error

The screen-space error (SSE) has been used in the literature to evaluate visual fidelity in model simplification. Balmelli used in [2] the peak signal-to-noise ratio (PSNR) of the rendered image as an objective measure of visual fidelity for simplified terrain models. For a general 3D model, because a single image cannot capture the entire appearance, Lindstrom et al. in [7] proposed to take virtual snapshots of the model from a number of different viewpoints around the object, and combine the image differences into a single measure of error. Mathematically, given two sets of $M \times N$ images, $\mathcal{Y} = \{Y_k\}$ and $\mathcal{Y}' = \{Y'_k\}$, $k = 1, ..., L$, the mean squared difference between these two sets of images is computed by

$$\sigma^2(\mathcal{Y}, \mathcal{Y}') = \frac{1}{LMN} \sum_{k=1}^{L} \sum_{j=1}^{M} \sum_{i=1}^{N} (y_{ijk} - y'_{ijk})^2. \qquad (1)$$

To ensure even coverage of image samples, the viewpoints need to be arranged so as to approximate a sphere of camera positions surrounding the object and to be (near) equidistant from each other. There are a number of possible configurations in practice. And one representative is to use the *small rhombicuboctahedron* as shown on the right, which was empirically exploited in [7] for off-line quality evaluation in order to perform image-driven mesh simplification. Hereafter, for evaluation of SSE for a rendered 3D model, we adopt the small rhombicuboctahedron and Equation (1) with screen-space resolution 512×512 pixels.

The screen-space error has been confirmed working successfully in evaluation of visual quality of multi-resolution meshes without texture mapping [14]. Nevertheless, it may not reflect the perceptual quality properly when we replace simplified mesh geometry with multi-resolution texture images. To demonstrate this fact, we run some experiments on the MANDRILL model (courtesy of Balmelli) shown in Figure 3a. We first generate a highly simplified mesh while keeping the texture at full resolution and the rendered right-side viewpoint is shown in Figure 3b. Then, we map a low resolution texture on the simplified geometry as shown in Figure 3c. When we compute the PSNR of the rendered images of the right-side viewpoint, surprisingly the simplified geometry with the full-resolution texture has lower PSNR value than the model with the low-resolution texture. Furthermore, when we compute the PSNR using the small rhombicuboctahedron, the difference in PSNR is very small whereas the visual difference is significant by looking at the two models. This observation reveals the existing inaccuracy of using SSE as the fidelity measure of jointly simplified textured models.



(a) 25,644 faces	(b) 788 faces	(c) 788 faces
24 bits/pixel	24 bits/pixel	0.05 bits/pixel
PSNR-r :	22.96 dB	23.23 dB
PSNR-s :	23.95 dB	23.80 dB

Figure 3: Captured images of the MANDRILL model. PSNR-r is the PSNR of the right-side viewpoint as shown above. PSNR-s is the average PSNR computed using the small rhombicuboctahedron.

Moreover, SSE is expensive for computation as rendering is a costly operation in many graphical systems. In Section 5, we will see that this hindrance becomes very severe when we apply this error metric to joint mesh and texture optimization, where a number of rendering operations are required in order to evaluate the quality of all pairs of meshes and textures. In addition, SSE is sensitive to the screen-space coordinates calculated during the rendering process. A slight change of the bounding box during simplification will result in small screen coordinate deviation. Due to texture mapping, such small deviation may incur significant increase in screen-space error even though no visual degradation can be perceived by human observers. All these factors make the usage of the screen-space error less attractive. In the next section, we present a quality measure that is fast, independent of the screen-space coordinates, and accurately captures the perceptual fidelity of the jointly simplified textured models.

4.2 FQM: A Fast Quality Measure

To effectively measure the perceptual fidelity of a textured model that is composed of a simplified mesh and a low-resolution texture, we need first to study the effect of geometry and texture simplification processes *separately*. Watson et al. in [14] showed that both the screen-space error and the surface distance successfully predict the perceptual fidelity in the simplified geometry if no texture is mapped. Similarly, the mean squared error (MSE) is successful in capturing the perceptual fidelity in 2D images. Driven by these observations, we propose to combine both of these measures to estimate the quality of the textured model. More specifically, we combine the mean squared texture deviation (MSD) explained in Section 3 and the mean squared error (MSE). Mathematically, we define the quality ($\mathcal{Q}$) of the textured model that is composed of mesh M_i and texture T_j as

$$\mathcal{Q} = (1 - \frac{MSD}{L^2})^{\lambda} \cdot (1 - \frac{MSE}{255^2})^{(1-\lambda)}, \tag{2}$$

where L is the diagonal of the bounding box; $\lambda \in [0, 1]$ is introduced as an *equalization* factor between the mesh and the texture, and will be discussed later in this section. MSD is the mean squared texture deviation of the mesh M_i, and MSE is the mean squared error of the texture T_j. Note that all quantities in (2) are available during the compression algorithm and only λ needs to be computed during the bit-allocation stage (Figure 1). In addition, $\mathcal{Q}$ does not depend on the viewpoint. Henceforth, this quality measure can be applied to general 3D objects including terrain models.

Taking the logarithm of both sides in (2), we rewrite $\mathcal{Q}$ as

$$\mathcal{Q} = \lambda \mathcal{Q}_{\mathcal{G}} + (1 - \lambda)\mathcal{Q}_{\mathcal{T}}, \tag{3}$$

where

$$\mathcal{Q}_{\mathcal{G}} = \log(1 - \frac{MSD}{L^2}), \text{ and } \mathcal{Q}_{\mathcal{T}} = \log(1 - \frac{MSE}{255^2}). \tag{4}$$

To simplify notation, we keep using $\mathcal{Q}$ in (3) to denote the quality measure.

The main objective of texture mapping is to replace complex geometry with the less expensive image to reflect certain details in the model. Therefore, one might expect the texture to be more important than the geometry in all cases. Nevertheless, the equalization factor, λ, is strongly dependent on the characteristics of the model, the spatial and spectral distribution of the texture image, the granularity of the triangular mesh, and the masking effect of substituting surface with texture. In general, the equalization factor should be expressed as a function of given resolutions of the mesh and the texture image, i.e., $\lambda = \lambda(M_i, T_j)$. For simplicity, in this paper we assume that the variation of λ among different resolutions is small and for a given textured model, we approximate the equalization factor with a constant value. The experimental results conformed the validity of this assumption.

Taking partial derivatives of Equation (3), we get

$$\lambda = \frac{\partial \mathcal{Q}}{\partial \mathcal{Q}_{\mathcal{G}}} = \frac{\partial \mathcal{Q}}{\partial \mathcal{R}_{\mathcal{G}}} / \frac{\partial \mathcal{Q}_{\mathcal{G}}}{\partial \mathcal{R}_{\mathcal{G}}}, \tag{5}$$

and

$$(1 - \lambda) = \frac{\partial \mathcal{Q}}{\partial \mathcal{Q}_{\mathcal{T}}} = \frac{\partial \mathcal{Q}}{\partial \mathcal{R}_{\mathcal{T}}} / \frac{\partial \mathcal{Q}_{\mathcal{T}}}{\partial \mathcal{R}_{\mathcal{T}}}, \tag{6}$$

where $\mathcal{R}_{\mathcal{G}}$ and $\mathcal{R}_{\mathcal{T}}$ are the bitrates of the mesh and texture, respectively. $(\mathcal{Q}_{\mathcal{G}}, \mathcal{R}_{\mathcal{G}})$ and $(\mathcal{Q}_{\mathcal{T}}, \mathcal{R}_{\mathcal{T}})$ are measured quantities in mesh and texture simplification processes, respectively. Combining Equations (5) and (6) into a single equation as

$$\frac{\lambda}{1 - \lambda} = \frac{\rho_{\mathcal{G}}}{\rho_{\mathcal{T}}} \Rightarrow \lambda = \frac{\rho_{\mathcal{G}}}{\rho_{\mathcal{G}} + \rho_{\mathcal{T}}}, \tag{7}$$

where

$$\rho_{\mathcal{G}} \triangleq \frac{\partial \mathcal{Q}}{\partial \mathcal{R}_{\mathcal{G}}} / \frac{\partial \mathcal{Q}_{\mathcal{G}}}{\partial \mathcal{R}_{\mathcal{G}}}, \text{ and } \rho_{\mathcal{T}} \triangleq \frac{\partial \mathcal{Q}}{\partial \mathcal{R}_{\mathcal{T}}} / \frac{\partial \mathcal{Q}_{\mathcal{T}}}{\partial \mathcal{R}_{\mathcal{T}}}.$$

We present a computational method of estimating λ based on (5)-(7) and two heuristics: (i) although not always, the screen-space error provides meaningful prediction of quality under certain conditions, and (ii) a relative-manner measure will actually be sufficient to evaluate visual quality of different pairs of mesh and texture resolutions. We carefully select the pairs to be rendered so as to obtain finely quantified quality difference among them.

We first normalize the scales of $\mathcal{Q}_{\mathcal{G}}$ and $\mathcal{Q}_{\mathcal{T}}$ in their respective coordinate systems so that the full resolutions of mesh and texture have measures of 1, and their coarsest versions have quality rated by 0. We believe that the scaling effect between two coordinate systems will be accounted for by the equalization factor and therefore it will not affect the eventual quality measures.

The LODs that we choose to render in the screen space will fall around the *turning* points on the curves of $(\mathcal{R}_{\mathcal{G}}, \mathcal{Q}_{\mathcal{G}})$ and $(\mathcal{R}_{\mathcal{T}}, \mathcal{Q}_{\mathcal{T}})$, respectively. Figure 4 is a demonstration of this selection conducted in $(\mathcal{R}_{\mathcal{G}}, \mathcal{Q}_{\mathcal{G}})$ coordinate system. In doing so, we find an index, k, such that the line constructed by two points, $({\mathcal{R}_{\mathcal{G}}}^{(k)}, {\mathcal{Q}_{\mathcal{G}}}^{(k)})$ and $({\mathcal{R}_{\mathcal{G}}}^{(k+1)}, {\mathcal{Q}_{\mathcal{G}}}^{(k+1)})$, makes a 45°-angle (or the nearest if not exact) with the x-axis. We denote

$$\Delta {\mathcal{Q}_{\mathcal{G}}}^{(k)} = {\mathcal{Q}_{\mathcal{G}}}^{(k)} - {\mathcal{Q}_{\mathcal{G}}}^{(k+1)}, \Delta {\mathcal{R}_{\mathcal{G}}}^{(k)} = {\mathcal{R}_{\mathcal{G}}}^{(k)} - {\mathcal{R}_{\mathcal{G}}}^{(k+1)}.$$

Likewise, we find an index, l, for the texture such that $({\mathcal{R}_{\mathcal{T}}}^{(l)}, {\mathcal{Q}_{\mathcal{T}}}^{(l)})$ is the *turning* point on the curve of $(\mathcal{R}_{\mathcal{T}}, \mathcal{Q}_{\mathcal{T}})$ and denote

$$\Delta {\mathcal{Q}_{\mathcal{T}}}^{(l)} = {\mathcal{Q}_{\mathcal{T}}}^{(l)} - {\mathcal{Q}_{\mathcal{T}}}^{(l+1)}, \Delta {\mathcal{R}_{\mathcal{T}}}^{(l)} = {\mathcal{R}_{\mathcal{T}}}^{(l)} - {\mathcal{R}_{\mathcal{T}}}^{(l+1)}.$$

By choosing the LODs around the turning points, we desire to obtain well bounded numerical values of $\frac{\Delta \mathcal{Q}_{\mathcal{G}}}{\Delta \mathcal{R}_{\mathcal{G}}}$ and $\frac{\Delta \mathcal{Q}_{\mathcal{T}}}{\Delta \mathcal{R}_{\mathcal{T}}}$ for the denominators of the right hand sides in Equation (5) and (6), hence minimizing potential arithmetic error. In addition, according to our observation, rendering the selected LODs provides meaningful measurements of the screen-space error, which lead to a good estimate of $\partial \mathcal{Q}$ in (5) and (6).

Assume we have selected M_k, M_{k+1} and T_l, T_{l+1} on respective curves. Three pairs of mesh and texture resolutions, (M_k, T_l), (M_{k+1}, T_l), and (M_k, T_{l+1}), as well as the coarsest

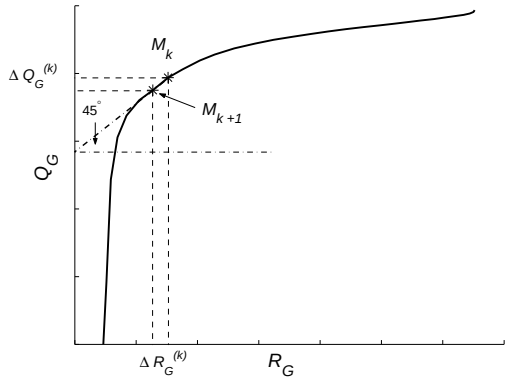


Figure 4: Illustration of selecting the *turning* points on $(\mathcal{R}_\mathcal{G}, \mathcal{Q}_\mathcal{G})$ curve: (M_k, M_{k+1}) constructs a line that has a 45°-angle with the x-axis.

representation, (M_n, T_m), are then rendered in the screen space and their errors are measured respectively. The corresponding numerical results are represented by $\sigma^2_{k,l}$, $\sigma^2_{k+1,l}$, $\sigma^2_{k,l+1}$, and $\sigma^2_{n,m}$. All the quantities are computed in the screen space as explained in Section 4.1.

We compute

$$\mathcal{P} = \log(1 - \frac{\sigma^2}{255^2})$$

for the measured mean squared errors and again normalize the results so that $\mathcal{P}_{n,m} = 0$ for the coarsest representation, (M_n, T_m), and consequently, $\mathcal{P}_{0,0} = 1$ for the full resolution model, (M_0, T_0). Denoting the corresponding results for (M_k, T_l), (M_{k+1}, T_l), and (M_k, T_{l+1}) by $\mathcal{P}_{k,l}$, $\mathcal{P}_{k+1,l}$, and $\mathcal{P}_{k,l+1}$, respectively, we estimate $\partial\mathcal{Q}$ in (5) and (6) by

$$\Delta\mathcal{Q}' = \mathcal{P}_{k,l} - \mathcal{P}_{k+1,l}, \text{ and } \Delta\mathcal{Q}'' = \mathcal{P}_{k,l} - \mathcal{P}_{k,l+1}.$$

Using (7), the equalization factor, λ, is finally computed by

$$\lambda = \frac{\Delta\mathcal{Q}'/\Delta{\mathcal{Q}_\mathcal{G}}^{(k)}}{\Delta\mathcal{Q}'/\Delta{\mathcal{Q}_\mathcal{G}}^{(k)} + \Delta\mathcal{Q}''/\Delta{\mathcal{Q}_\mathcal{T}}^{(l)}}. \quad (8)$$

Note that $\Delta{\mathcal{R}_\mathcal{G}}^{(k)}$ and $\Delta{\mathcal{R}_\mathcal{T}}^{(l)}$ have been cancelled in (8) as common factors.

The above process can be repeated by choosing more sample pairs in the screen space. For instance, in our experiments presented in Section 6, we choose to render two more pairs, (M_{k-1}, T_l) and (M_k, T_{l-1}), and follow the computation but using (M_{k-1}, T_l), (M_k, T_{l-1}) together with (M_k, T_l) and (M_n, T_m). We then have the average as the final estimate of λ. Combining (3) with (8) gives the complete expression of the proposed quality measure(FQM). Clearly, FQM is computationally efficient as only few measurements are required in the screen space. In addition, the experimental results presented in Section 6 verify that FQM successfully captures the visual fidelity of jointly simplified models, although it is not a rigorously proven metric.

5. OPTIMAL BIT ALLOCATION

We now apply the proposed visual fidelity measure (FQM) to the bit allocation block in Figure 1. The bit-allocation algorithm searches for the proper combination of mesh and texture resolutions that maximizes the quality measured by FQM for a given bit budget. The mathematical formulation of this rate-distortion optimization framework can be stated as follows: Given a bitrate constraint, $\mathcal{C}$, and the distortion (quality) measure, $\mathcal{Q}$, the best representation of the 3D data achieved with the set of meshes, $\{M_i\}_{i=0\ldots n}$, and the set of textures, $\{T_j\}_{j=0\ldots m}$, is given by

$$(M_k, T_l)_{opt} = \underset{(i,j):\mathcal{R}_\mathcal{G}(M_i)+\mathcal{R}_\mathcal{T}(T_j)\leq\mathcal{C}}{\arg\max} \mathcal{Q}(M_i, T_j), \quad (9)$$

where $\mathcal{R}_\mathcal{G}(M_i)$ and $\mathcal{R}_\mathcal{T}(T_j)$ denote the bitrates of the compressed mesh M_i and the texture T_j, respectively; $\mathcal{Q}(M_i, T_j)$ is the quality of the jointly simplified model, evaluated using FQM discussed in Section 4.2.

The optimal solution of (9) can be obtained by exhaustive search over the space of solutions, i.e., comparing all possible pairs of mesh and texture resolutions satisfying the bit budget. Obviously, this process has computational complexity of $O(n \times m)$. Balmelli in [2] showed that using a greedy approach to solve the optimization problem results in a solution close to the optimal solution obtained via the extensive search. In both methods, using SSE as the quality measure will make the search infeasible. In contrast, the computation of FQM is affordable as only numerical comparisons will be conducted with a small (and constant) overhead for estimating the equalization factor. This is because of the simple structure of FQM. For the space of solutions that has dimensions 40×30, for example, while computing FQM requires virtually capturing $4 \times 24 = 96$ screen images (with least selected samples for rendering), using SSE as distortion measure will impose a total number of $24 \times 40 \times 30 = 28,800$ rendering operations.

For the applications that deal with large 3D models or require precise bit allocation between the mesh and the texture, the space of solutions may get substantially large. Thus the exhaustive search becomes inefficient and heuristic methods are desirable. One of such heuristics is the marginal analysis introduced in [2] using a resource reduction approach, which searches greedily for the best tradeoff while reducing the resolutions of the mesh and the texture. In other words, at each step either the mesh bit rate or the texture bit rate is reduced such that the decrease in distortion is minimal.

The marginal analysis gains a linear computational complexity at the price of providing suboptimal solutions. Instead, we exploit another heuristic with linear complexity, innovated from the decoupled structure of the FQM and its monotonicity as a function of the resolutions of mesh and texture. With this particular distortion function, instead of performing exhaustive search over the space of solutions, we expect to find the optimal operating point on the boundary of the feasible region, namely,

$$(M_k, T_l)_{opt} = \underset{(i,j):(M_i,T_j)\in\mathcal{B}_\mathcal{C}}{\arg\max} \mathcal{Q}(M_i, T_j), \quad (10)$$

where $\mathcal{B}_\mathcal{C}$ denotes the boundary of the feasible region determined by the bit budget $\mathcal{C}$. It is easy to prove that $|\mathcal{B}_\mathcal{C}| = \min(n, m)$, and the procedure of finding the boundary has computational complexity of $O(n+m)$. That is, the computation complexity of (10) is linear with respect to the decoupled dimensions of the solutions space. Further investigation on the efficiency of the above heuristics as well as the marginal analysis will be conducted in Section 6 based on our experiments.

6. EXPERIMENTAL RESULTS

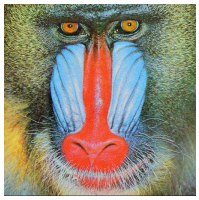
We have tested the proposed quality measure on different textured models and in here we report the results for

two models: the MANDRILL model[1] and the ZEBRA model[2]. Both models have 24-bit colored texture images and the texture coordinates associated with mesh vertices are quantized using 10-bit quantizers. For the mesh geometry, 10-bit quantizers are used for MANDRILL (25,644 faces) while 12-bit quantizers are used for ZEBRA (58,494 faces). Using the TD-CPM and the SPIHT algorithms, we generate 30 LODs for the MANDRILL mesh and 20 resolutions for the MANDRILL texture (coding rate 0.05-1.0 bits/pixel). Similarly, we generate 40 LODs for the ZEBRA mesh and 30 resolutions for the for ZEBRA texture (0.01-0.4 bits/pixel).

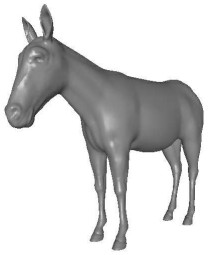
6.1 Objective Results

Following the algorithm described in Section 4.2, we found the equalization factor for MANDRLL and ZEBRA models to be 0.40 and 0.64, respectively. Note that λ is smaller than 0.5 for MANDRILL whereas λ is greater than 0.5 for ZEBRA. Intuitively, such relations of the *equalization* factor imply that for the MANDRILL model, the distortion of texture has larger impact than geometry on the visual fidelity of approximated models. In contrast, the impact of the texture error on the quality is less than geometry when we construct a simplified representation of ZEBRA. This conclusion can be conformed by subjective comparison of the two models (refer to Figures 7 and 8). The MANDRILL texture image has rich color elements and has fine details that are necessary for the model appearance, while the the ZEBRA texture has fewer details (see Figure 5).

25,644 faces
(a) MANDRILL mesh



512 × 512 pixels
(b) MANDRILL texture

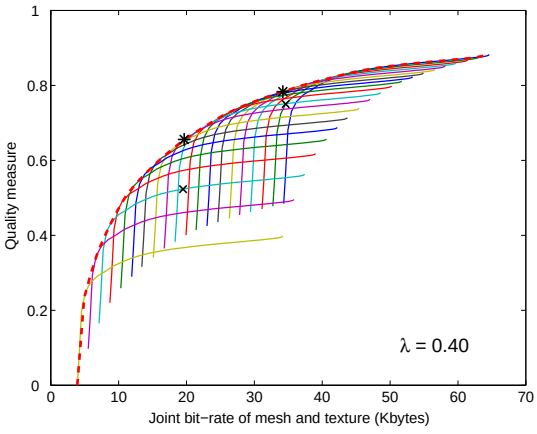


58,494 faces
(c) ZEBRA mesh

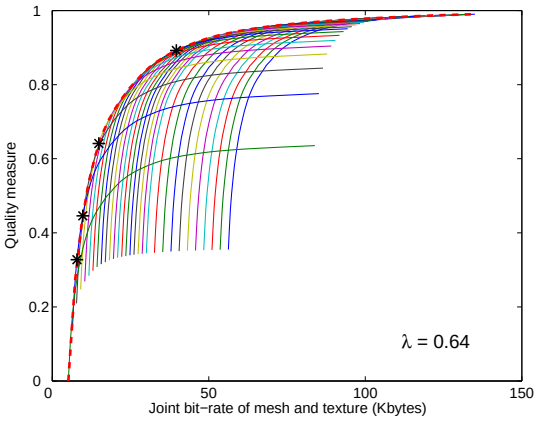
1024 × 1024 pixels
(d) ZEBRA texture

Figure 5: The geometry and the texture of the MANDRILL and the ZEBRA models.

Figures 6 (a) and (b) depict the calculated FQM measure of different mesh and texture pairs (M_i and T_j) of the MANDRILL and the ZEBRA models, respectively. The x-axis in these plots is the total bit budget. Each thin curve in the plots represents a constant rate for the texture with increasing mesh bitrates. In each plot, the dashed (red) line shows the optimal rate-distortion envelope of the convex hull, which is the upper bound of the visual quality that could be achieved under all configurations.



(a) MANDRILL



(b) ZEBRA

Figure 6: Rate-distortion curves for the MANDRILL and the ZEBRA models. Each thin curve corresponds to a fixed texture rate. The dashed red envelope shows the optimal rate-distortion pairs and gives the best representations under limited bit budget.

We observe in Figure 6(a) that for the same mesh resolution, the quality increases considerably as the texture resolution (or the bitrate, equivalently) increases. In comparison, the experiment presented in Figure 3 showed that SSE fails in reflecting the perceptual fidelity when mapping textures with different resolutions to coarse mesh geometry[3]. As anticipated, FQM measures the perceptual difference more properly than SSE.

In the plot of ZEBRA, the increase in the quality as the texture rate increases for a fixed mesh level is not as significant as in the MANDRILL case (compare the plots in Figure 6). Instead, the improvement quickly decreases and after certain resolution, increasing texture rate does not result in noticeable difference in the total quality. This observation is actually not surprising as from the value of λ, we expect that the texture error of this model has less impact on the visual fidelity compared to the MANDRILL. This is inline with our anticipation that the equalization factor, λ, and consequently the proposed quality measure, $\mathcal{Q}$, mean-

[1] courtesy of L. Balmelli
[2] courtesy of P. Lindstrom

[3] In Figure 6, the lowest points of the thin curves correspond to mapping the textures with different resolutions to the coarsest mesh geometry.

ingfully describe the unequal impacts of the geometry and the texture resolutions on the perceptual quality of texture models.

6.2 Subjective Results

Figures 7 and 8 provide subjective evaluation for the MANDRILL and the ZEBRA models. In Figures 7 (a)-(d), we present two pairs of captured images, with approximations made under bit budgets 20KB and 35KB, respectively. Figures 7 (a) and (c) show the optimal pairs found by exhaustive search. The corresponding points are marked in Figure 6 (a) with asterisks on the envelope and crosses for the rest. Notice the improvement in the visual quality as we compare (b) and (a). For example, the forehead area in (b) is more blurred than the one in (a). Figures 7 (c) and (d) have smaller difference in quality for a given bit budget of 35KB. Higher distortion is noticeable in Figure 7(d) around the cheek and the chin area when viewing the model from the right side.

Similarly, for the ZEBRA model, Figures 8 (a)-(d) capture the rendered results of a few samples on the envelope of Figure 6 (b) while the bit budget decreases from 40KB to 8KB. Note how the feet and the stripes gradually distort while becoming more and more vague from Figure 8(a) to Figure 8(d) as the bit rate decreases. All the subjective comparisons are consistent with the FQM measures, showing that FQM successfully captures the visual fidelity of jointly simplified models.

6.3 Efficiency of the Bit-Allocation Algorithm

We now investigate the bit-allocation framework introduced in Section 5. In Table 1, we present a comparative study on the two algorithms, *boundary search* and *marginal analysis*, using the 1,200 pairs of ZEBRA models generated previously. For a sequence of bit budgets varying from 8KB to 40KB, the quality measured using the boundary search (Q_B) and the quality measured using the marginal analysis (Q_M) are listed in Table 1 along with the corresponding bit-allocation percentages of the mesh and the texture. The boundary search algorithm always gives the maximum quality measures under corresponding bit constraints, i.e., $\mathcal{Q}_B = \mathcal{Q}_{opt}$. Compared to the boundary search, the marginal analysis algorithm is suboptimal. The difference between the marginal analysis and the optima given by the boundary search is highest at low bit budgets.

Figures 9 and 10 visually compare these two methods. In Figure 9, the blue line with plus signs shows the execution path of the marginal analysis with bit budget 15KB, while the black line with points indicates the path of the boundary search. The solutions are marked by circles on the contour. The corresponding quantities are found in Table 1 to be $\mathcal{Q}_B = \mathcal{Q}_{opt} = 0.6410$ and $\mathcal{Q}_M = 0.6209$, respectively. As can be seen from Figure 10, even though the marginal analysis provides a near-optimal solution, perceptual difference in visual fidelity is still noticeable from certain perspective when we subjectively compare the results.

Table 1: Comparison of the boundary search and the marginal analysis algorithms (Bit-allocation percentages are listed following the quality measures for the mesh and the texture, respectively).

$\mathcal{C}$ (KB)	$\mathcal{Q}_B$ (Boundary)	$\mathcal{Q}_M$ (Marginal)
8	0.3275 (67%, 33%)	0.2092 (49%, 51%)
10	0.4459 (73%, 27%)	0.3605 (55%, 45%)
12	0.5395 (66%, 34%)	0.5258 (56%, 44%)
15	0.6410 (65%, 35%)	0.6209 (63%, 37%)
18	0.6978 (63%, 37%)	0.6978 (63%, 37%)
22	0.7555 (64%, 36%)	0.7555 (64%, 36%)
26	0.7953 (64%, 36%)	0.7931 (69%, 31%)
30	0.8355 (69%, 31%)	0.8355 (69%, 31%)
37	0.8766 (67%, 33%)	0.8690 (70%, 30%)
40	0.8922 (67%, 33%)	0.8880 (73%, 27%)

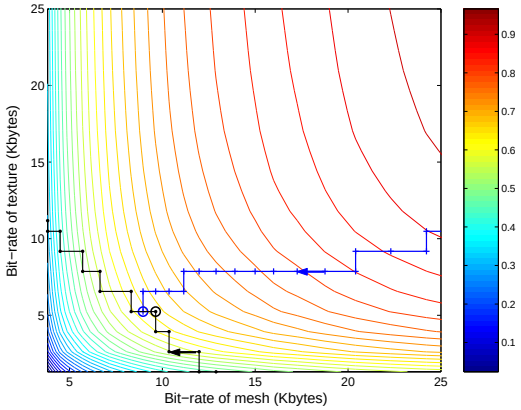


Figure 9: Execution paths of the boundary search (dotted black line) and the marginal analysis algorithms (blue line with "+" marks) when the bit budget is 15KB.

7. CONCLUSIONS

We have proposed a novel quality measure of visual fidelity for multi-resolution textured surfaces. The proposed quality measure (FQM) combines the error on mesh geometry and the distortion of texture image through an equalization factor, which is estimated in the rendering space. Using FQM, we presented a bit-allocation algorithm to transmit textured models over bandwidth-limited channels. The bit budget is optimally distributed between the mesh and the texture to achieve the maximum quality. Our experiments showed that the proposed system is both effective and efficient.

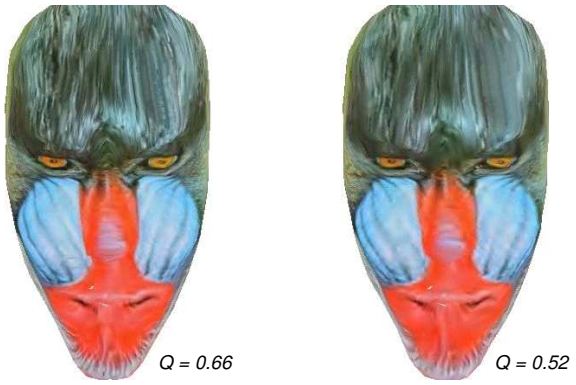
Although we used sampled screen-space error to compute the equalization factor in this paper, the FQM is essentially independent of the rendering space and can work with any other methods that well estimate the equalization factor. In addition, the mean squared texture deviation (MSD) and the mean squared image error (MSE) were employed as the measures of geometry distance and the texture distortion, respectively, to construct the quality function. It is worth to be pointed out, however, that the framework in the paper is general enough that any metrics of surface distance and image distortion can be used, although with potentially different computation complexity.

8. REFERENCES

[1] M. J. Abasolo and F. Perales. Geometric-textured bitree: Transmission of a multiresolution terrain across the internet. *Journal of Computer Science and Technology*, 2(7):34–41, October 2002.

[2] L. Balmelli. Rate-distortion optimal mesh simplification for communications. PhD dissertation No 2260, Ecole Polytechnique Federale de Lausanne, Switzerland, 2001.

[3] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. In *Proc. Eurographics*, volume 17(2), pages 167–174, June 1998.



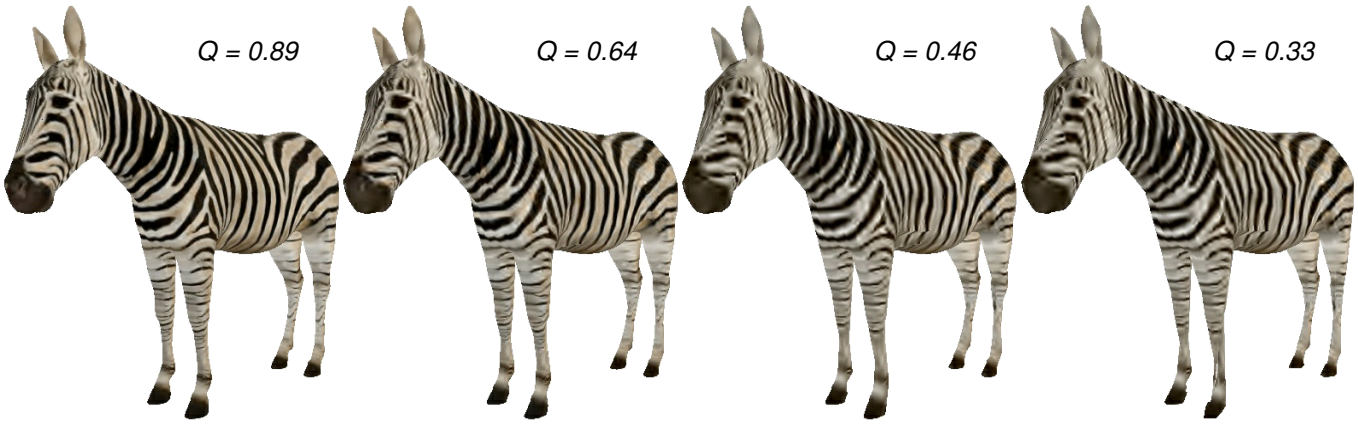
$(R_{\mathcal{G}}, R_{\mathcal{T}}) = (5.26, 14.4)$ KB (2444 faces, 0.45 bits/pixel) (a)

$(R_{\mathcal{G}}, R_{\mathcal{T}}) = (14.7, 4.79)$ KB (7644 faces, 0.15 bits/pixel) (b)

$(R_{\mathcal{G}}, R_{\mathcal{T}}) = (10.2, 24.0)$ KB (4844 faces, 0.75 bits/pixel) (c)

$(R_{\mathcal{G}}, R_{\mathcal{T}}) = (4.22, 30.4)$ KB (2044 faces, 0.95 bits/pixel) (d)

Figure 7: Subjective evaluation of the approximated MANDRILL model. Bit constraints for (a-b) and (c-d) are 20KB and 35KB, respectively. (a) and (c) give the optimal mesh-texture pairs for different bit budgets.



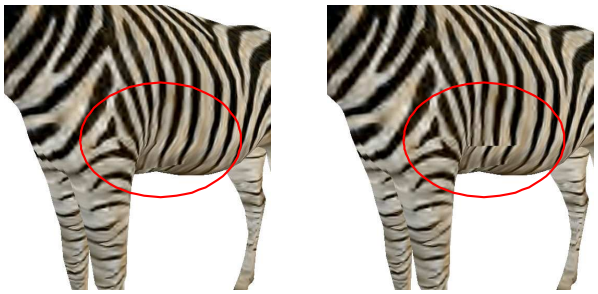
$(R_{\mathcal{G}}, R_{\mathcal{T}}) = (26.6, 13.1)$ KB (11806 faces, 0.09 bits/pixel) (a)

$(R_{\mathcal{G}}, R_{\mathcal{T}}) = (9.64, 5.24)$ KB (3370 faces, 0.04 bits/pixel) (b)

$(R_{\mathcal{G}}, R_{\mathcal{T}}) = (7.15, 2.62)$ KB (2330 faces, 0.02 bits/pixel) (c)

$(R_{\mathcal{G}}, R_{\mathcal{T}}) = (5.27, 2.62)$ KB (1586 faces, 0.02 bits/pixel) (d)

Figure 8: Subjective evaluation of the ZEBRA model, sampled along the envelope of Figure 6(b). The corresponding bit constraints are 40KB, 15KB, 10KB and 8KB, respectively.



Boundary search $\mathcal{Q}_{opt} = 0.6410$ (a)

Marginal analysis $\mathcal{Q}_{M} = 0.6209$ (b)

Figure 10: Comparison of two ZEBRA models obtained using (a) the boundary search and (b) the marginal analysis for a bit budget of 15KB.

[4] J. Cohen, M. Olano, and D. Manocha. Appearance-preserving simplification. In *Proc. ACM SIGGRAPH*, pages 115–122, 1998.

[5] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Proc. ACM SIGGRAPH*, pages 209–216, 1997.

[6] H. Hoppe. Progressive meshes. In *Proc. ACM SIGGRAPH*, pages 99–108, 1996.

[7] P. Lindstrom. Model simplification using image and geometry-based metrics. Ph.D. Dissertation, Georgia Institute of Technology, November 2000, 2000.

[8] P. Lindstrom and G. Turk. Fast and memory efficient polygonal simplification. In *Proc. IEEE Visualization*, pages 27–9–286, 1998.

[9] M. Okuda and T. Chen. Joint geometry/texture progressive coding of 3d models. In *Proc. IEEE Int. Conf. on Image Processing*, pages 632–635, September 2000.

[10] R. Pajarola and J. Rossignac. Compressed progressive meshes. *IEEE Trans. Visualization and Computer Graphics*, 6(1):79–93, January-March 2000.

[11] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Tech.*, 6(3):243–250, June 1996.

[12] P. Sander, J. Snyder, S. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *Proc. ACM SIGGRAPH*, pages 409–416, 2001.

[13] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Trans. Graphics*, 17(2):84–115, April 1998.

[14] B. Watson, A. Friedman, and A. McGaffey. Measuring and predicting visual fidelity. In *Proc. ACM SIGGRAPH*, pages 213–220, 2001.