Perceptually Guided Polygon Reduction

Lijun Qu and Gary W. Meyer

Abstract—The properties of the human visual system are taken into account, along with the geometric aspects of an object, in a new surface remeshing algorithm and a new mesh simplification algorithm. Both algorithms have a preprocessing step and are followed by the remeshing or mesh simplification steps. The preprocessing step computes an importance map that indicates the visual masking potential of the visual patterns on the surface. The importance map is then used to guide the remeshing or mesh simplification algorithms. Two different methods are proposed for computing an importance map that indicates the masking potential of the visual patterns on the surface. The first one is based on the Sarnoff visual discrimination metric, and the second one is inspired by the visual masking tool available in the current JPEG2000 standard. Given an importance map, the surface remeshing algorithm automatically distributes few samples to surface regions with strong visual masking properties due to surface texturing, lighting variations, bump mapping, surface reflectance, and interreflections. Similarly, the mesh simplification algorithm simplifies more aggressively where the light field of an object can hide more geometric artifacts.

Index Terms—Perceptually guided rendering, visual masking, visual perception, surface remeshing, simplification, level of detail.

1 Introduction

Texture maps, bump maps, environment maps, and surface reflections all have a dramatic impact on the appearance of a polygon mesh. Today, these *surface signals* are used to produce striking visual effects at little cost by employing the texture mapping and pixel shading hardware available on PC graphics cards. There has been a considerable amount of work in the field of computer graphics on the creation, processing, and usage of surface signals.

Surface signals have also been employed to develop perceptually based global illumination algorithms [35], to compress the texture map by allocating less texture space for low-frequency texture regions [4], and to generate a specialized signal parameterization [36] by taking into account the frequency distribution of the surface signal. However, very little work in the area of geometric modeling has taken surface signals into consideration.

Some researchers have noticed that surface signals can be useful in the area of geometric modeling. Ferwerda et al. [12] developed a visual masking model for computer graphics and observed that visual masking can have an impact on the representation of geometric models. The existence of surface signals on a geometric model can raise the visibility threshold of the geometric error due to surface tessellation. This elevated threshold should be taken into account by algorithms such as mesh simplification and surface remeshing algorithms that aim to determine a simplified representation for a geometric model. Fig. 1 shows a flat shaded cylinder with and without texture. Faceting artifacts can be seen clearly in Fig. 1a, but no faceting artifacts can be seen in Fig. 1b. For the cylinder with texture shown in Fig. 1b, a more coarse geometric representation would suffice.

In this paper, we propose a surface remeshing algorithm [31] and a mesh simplification algorithm that take into account the surface signals on the mesh. We are particularly interested in the perceptual properties of the surface signals. Most surface remeshing or mesh simplification algorithms distribute samples on the surface according to the geometric properties of the mesh such as surface curvature. In this paper, the distribution of samples is guided both by the geometric properties of the mesh as well as the perceptual properties of the surface signals.

This paper makes contributions in the following areas:

- We extend the current state of the art in perceptually based level of detail algorithms to include visual masking. Visual masking requires multiscale and multiorientation decomposition of the signal and is difficult to apply in surface remeshing and simplification algorithms.
- We propose two methods that compute the perceptual properties of the surface signal. The first one is based on the Sarnoff visual discrimination metric (VDM). The second one is based on the visual masking tool recently developed for the JPEG2000 standard.
- We develop a surface remeshing and a mesh simplification algorithm that take into account both the geometric properties and the perceptual properties of the surface signals on the mesh.

The remainder of this paper is organized as follows: Section 2 reviews some related work in perceptually guided rendering, perceptually guided level of detail, surface remeshing and mesh simplifications, and the interaction between surface signals and geometry. In Section 3, we introduce some background on the properties of the human visual system. We then describe two algorithms that compute the visual masking potential of a texture in

- *The authors are with the Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CS Building, 200 Union Street SE, Minneapolis, MN 55455. E-mail: {lijun, meyer}@cs.umn.edu.*

Manuscript received 9 Aug. 2007; revised 14 Feb. 2008; accepted 21 Feb. 2008; published online 12 March 2008.
Recommended for acceptance by B. Watson.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2007-08-0104.
Digital Object Identifier no. 10.1109/TVCG.2008.51.

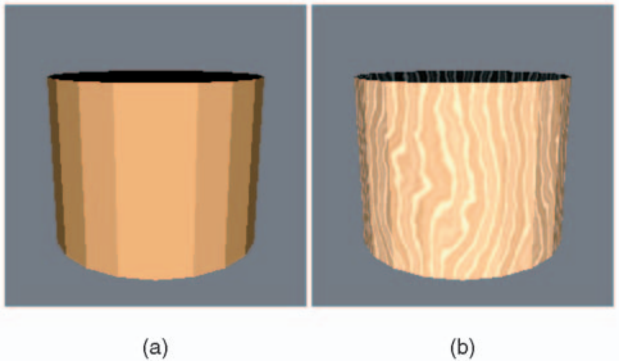


Fig. 1. Flat shaded cylinder (a) without and (b) with texture (after Ferwerda et al. [12]).

Section 4. In Sections 5 and 6, we describe in detail our perceptually guided remeshing and simplification algorithms. In Section 7, we discuss our perceptually guided approach to polygon reduction and compare it to some previous work in this area. Finally, we summarize our work in Section 8.

2 Previous Work

In this section, we present some previous work in the area of perceptually guided rendering, perceptually guided level of detail, surface remeshing and mesh simplification algorithms, and the relationship between the surface signal and geometry.

2.1 Perceptually Guided Rendering

Visual perception has traditionally been exploited in computer graphics to accelerate expensive global illumination algorithms. Please refer to a recent report for a summary [29]. By taking advantage of properties of the human visual system, computational resources can be intelligently allocated where they are most needed. Furthermore, visual perception can serve as a stopping condition for image synthesis algorithms. Due to the characteristics of the human visual system, image synthesis algorithms can halt at a point that yields consistent image quality. Several models of the human visual system have been used in computer graphics. Bolin and Meyer [5] propose an adaptive sampling ray tracer guided by a modified Sarnoff VDM [27] that samples the image plane based on the image content in the rendered images. Volevich et al. [42] use the visible differences predictor [10] to improve the performance of progressive global illumination algorithms. Ramanarayanan et al. [34] introduce the concept of visual equivalence and propose visual equivalence predictors (VEPs). They also show that VEPs can be used to improve the efficiency of rendering algorithms. Other schemes have also been proposed to speed up image synthesis by taking advantage of the human visual system [35], [38], [44]. Our algorithm shares similarities with these algorithms but they are applied to a different problem domain.

2.2 Perceptually Guided Level of Detail

The image-driven simplification framework developed by Lindstrom and Turk [25] departs from approaches that make polygon simplification decisions based on geometric error. Instead, they use the root mean square image difference metric. Their approach prioritizes edge collapse operations based on the root mean square difference in pictures created before and after the deletion takes place. They demonstrate that image-driven simplification can produce results better than or equal to most geometry-based mesh simplification algorithms. Lindstrom [24] employs a perceptually motivated metric in a mesh simplification algorithm. Luebke and Hallen [28] propose perceptually driven mesh simplification that controls the simplification using psychophysical models of visual perception. They map an edge collapse operation to the worst contrast grating introduced by the edge in question. They later extended their work to the simplification of lit textured meshes [47]. However, visual masking is not included. Our paper has the same goal as their research. However, we use a contemporary model of the human visual system, which includes threshold-versus-intensity, contrast sensitivity, and visual masking. More recently, Lee et al. [21] introduce the idea of mesh saliency as a measure of regional importance for geometric models and then integrate this information into a mesh simplification algorithm. User studies on mesh saliency [17], [18] have shown its effectiveness for mesh simplification algorithms. User studies have also been conducted to measure and predict the visual fidelity of level of detail models [46].

2.3 Surface Remeshing and Simplification

With the advance of model acquisition techniques, there has been a considerable amount of work in the area of surface remeshing and mesh simplification. We only list publications most significant to our own work. Alliez et al. [3] propose a novel interactive technique that first partitions the model into patches homeomorphic to disks, and then parameterizes each patch over a planar domain. Most of the remeshing operations can then be performed in the 2D parametric domain instead of 3D. In their recent work on anisotropic remeshing [1], they show that sampling along the principle curvature directions can produce compactly represented meshes. Some researchers have taken another approach to surface remeshing by working directly on the 3D mesh. Turk [41] designs an elegant algorithm that positions vertices by point repulsion. More recent work [39] employs a series of local operations to improve mesh quality. As for related mesh simplification algorithms, visibility-guided simplification by Zhang and Turk [50] simplifies a polygonal mesh by exploiting the fact that different parts of the model can have different visibility due to self-occlusions. Parts of the surface that have less chance to be seen can be simplified more aggressively than other portions of the surface. Kho and Garland [19] develop a mesh simplification algorithm that takes into account an importance map drawn by a user on the geometric model. The importance map can include semantic information in general not available to the algorithm without user intervention. One example is the eye portion of a humanoid model. Pojar and Schmalstieg [30] also develop a user-guided simplification algorithm. Mesh saliency [21] is considered in a simplification algorithm, where salient parts of the geometric model are simplified less aggressively than other parts of the model. In this paper, we present two polygon reduction algorithms that take into account the properties of the human visual system, especially visual masking.

2.4 Surface Signals and Geometry

In this article, surface signals are any variation of object lightness and color that is generated by mechanisms independent of the underlying geometry (diffuse textures, normal maps, environment maps, etc). Nowadays, most meshes come with surface signals. However, the majority of existing work either considers the problem of geometry simplification without taking into account the perceptual properties of the surface signals, or attacks the problem of construction, manipulation, and optimization of surface signals without considering the geometry. Until recently, there has been little work that explores the interaction of geometry and surface signal. Sander et al. [36] and Tewari et al. [40] propose signal-specialized surface parameterization that minimizes the signal stretch instead of the usual geometry stretch and show that the signal-specialized parameterization can improve the image quality due to less texture stretch. Carr and Hart [6] design an interactive painting system that dynamically adjusts the parameterization of the geometry according to the frequency content of the texture painted on the surface. Their system allocates more texture space to high-frequency texture regions, thus preserving the details of the texture during rendering. Cheng and Boulanger [7] propose a method for adaptive transmission of 3D textured models online called TexMesh, which reduces the resolution of both the geometry and the texture using scale-space and visual perception analysis in order to satisfy a given transmission bandwidth constraint. In this present article, we take advantage of the masking effects introduced by the surface signals to further reduce the number of polygons without causing noticeable visual artifacts.

3 Models of Human Visual System

Several models of human vision [10], [27] have been developed in the image science industry to assist in the creation of visual displays, image compression algorithms, and other imaging systems. Ferwerda et al. [12] proposed a model of human vision for computer graphics. In these models, the information processing of the human visual system generally consists of three stages: threshold-versus-intensity function, contrast sensitivity function, and visual masking. The threshold-versus-intensity function describes the nonlinear response of the visual system responding to different intensity levels. The contrast sensitivity function indicates how the visual system reacts to stimuli of different frequencies. The visual masking function describes the visibility of a foreground object when seen against some background.

In this work, we are specially interested in the visual masking properties of the human visual system. In a visual masking experiment [22], the foreground is called the signal, and the background is called the masker. The sum of the masker and the signal is shown to an observer. The contrast of the signal is increased until it approaches the visibility threshold of the observer. Fig. 2 shows the effect of visual masking. The abscissa represents the contrast of the masker, and the ordinate represents the contrast of the signal. As can be observed in the diagram, when the contrast of the masker is low, there is little or no masking. Masking becomes apparent as the contrast of the masker increases.

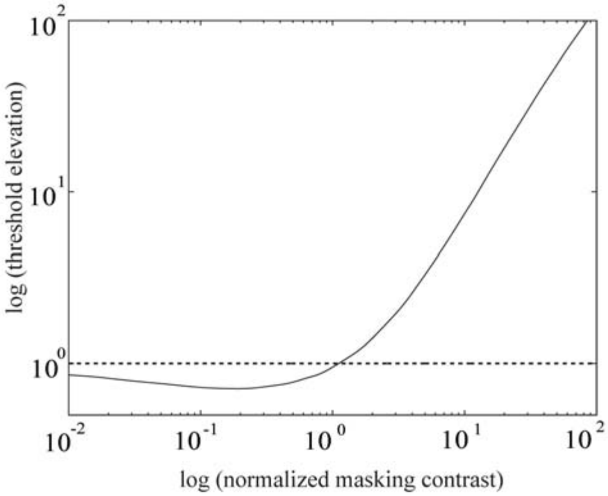


Fig. 2. The visual masking function of the human visual system, adapted from Ramasubramanian et al. [35].

Fig. 1 shows an example of visual masking. These two images are rendered with the same geometry. However, Fig. 1a shows obvious faceting artifacts, but Fig. 1b does not. This is due to the masking properties of the human visual system: the wood grain texture hides the faceting artifacts present in the second image. This phenomenon has been used in the image science industry for a long time. Artifacts introduced into the original image can be considered as the signal, and the original image is considered as the masker. For example, visual masking has been used in image compression algorithms [45]. Quantization can be performed more aggressively in areas of strong frequency content without introducing visual artifacts. The quantization noise introduced by aggressive quantization can be masked by the image content.

4 Importance Map Evaluation

In this paper, we want to compute the visual masking properties of a texture and use the results of this computation to guide a surface remeshing algorithm and a mesh simplification algorithm. Since the masking potential of a texture correlates strongly with the spatial frequency, contrast, and orientation of the test stimulus, any visual masking computation is not theoretically correct without considering the test stimulus itself. However, a well-designed algorithm based on models of the human visual system can still provide valuable information about the visual masking potential of a texture. Walter et al. [44] compute the visual masking properties of a texture using aspects of the JPEG image compression standard. Ramasubramanian et al. [35] propose a novel method to compute the visual masking properties of a texture by handling the luminance-dependent processing and spatially dependent processing separately and then combining them in an appropriate manner. In this section, we describe two algorithms that compute the

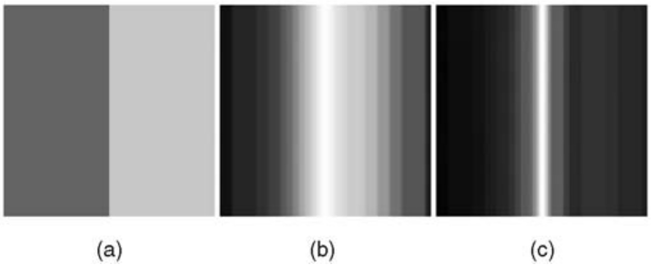


Fig. 3. (a) Is the original texture. (b) Is the visual masking map computed using the DC component of the original texture as the second comparison image. (c) Is the visual masking map using low-pass filtering of the original texture as the second comparison image.

masking potential of a texture. The first one is based on the Sarnoff VDM, and the second one is based on the visual masking tool developed in the JPEG2000 standard.

4.1 Algorithm 1—Sarnoff Visual Discrimination Metric

We propose to compute the visual masking properties of a texture using the Sarnoff VDM [27]. This allows us to take advantage of the accumulated experience and robustness that is built into this metric. Since the Sarnoff VDM takes two images as input, we need to have a second comparison image to feed in as input along with the original texture. Some researchers have tried novel ways to derive the second image or both images. Bolin and Meyer [5] determine two candidate images while ray tracing by using current estimates of the mean value and variance at each pixel. Volevich et al. [42] employ two intermediate global illumination solutions as input to the VDM.

According to Fourier theory, a texture can be decomposed into multiple frequencies. Since any nonzero frequency signal can potentially cause visual masking, we can remove all nonzero frequencies from the original texture which creates a uniform image whose intensity is the average of the original texture. This is also called the DC component of the original image. We then compare this image to the original texture. Since the Sarnoff VDM employs contemporary models of the human visual system, given the original texture and the DC component of the texture, the Sarnoff VDM will pick out visual differences for any nonzero frequencies in the original texture.

This approach would work if the original texture had similar intensity values across the texture. However, in general, this is not true for real-world textures, which have very nonuniform intensities. Regions with different intensities will be averaged together and they cannot be handled well by this approach. To solve this problem, some low frequencies in the original texture are allowed in the second comparison image to preserve the local average of the texture. Allowing some low frequencies in the second comparison image does not cause significant error in the final visual masking map because frequencies close to zero have relatively weaker visual masking compared to higher frequencies. Fig. 3 shows the visual masking caused by frequencies without considering the threshold-versus-intensity function of the human visual system. Notice that Fig. 3a correctly shows the visual masking caused by the step function in the original texture, while Fig. 3b incorrectly shows visual masking occurring across almost the entire texture.

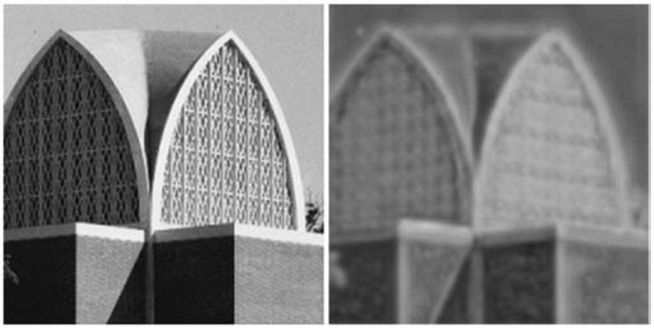


Fig. 4. The chapel image and its importance map calculated using the method derived from the Sarnoff VDM.

This can be implemented efficiently by low-pass filtering the original texture. In our implementation, we have used a Gaussian filter to remove most of the high frequencies. It is important to choose the right filter kernel size to filter the texture. If the filter only removes a small portion of the high frequencies in the original image, the visual masking caused by those frequencies left out in the second image will not show up in the final visual masking map. Therefore, the visual masking caused by those frequencies cannot be utilized in the remeshing algorithm. On the other hand, if the filter removes too many of the frequencies, the problem shown in Fig. 3b will occur. The aforementioned problem has more chance to happen if the image contains irregular intensity regions and these regions have sharp boundaries (see Fig. 3a). Hence, the optimal kernel size is a function of image content. In our implementation, we have experimentally selected a filter kernel of size 15 (1 degree of vision for our viewing distance and display dot pitch), and it works well for all the examples shown in this paper.

The threshold-versus-intensity function gives the error detection threshold corresponding to a given luminance background. To compute the error threshold described by the threshold-versus-intensity function, we have used the piecewise approximation developed by Ferwerda et al. [11]. To get the final visual masking map, we use a linear combination of these two maps. Note that we combine the results differently from Ramasubramanian et al. [35] since they compute an elevation map in the second step (which, in our case, is the just noticeable difference (JND) map, a kind of error threshold). Fig. 4 shows a chapel image and the final visual masking map generated by our algorithm. Notice that the window of the chapel shows stronger potential for visual masking while the background shows less possibility of visual masking. In addition, the right window shows visual masking is more likely than the left window because it has a higher base luminance level.

Since the Sarnoff VDM has been designed for physiological plausibility and has been verified by a number of applications, our approach is simple but has a strong underlying foundation.

4.2 Algorithm 2—JPEG2000

In this section, we propose another algorithm inspired by the visual optimization tools in the JPEG2000 standard that computes the visual masking potential of a texture. This algorithm has been applied successfully in a point-based rendering system [33].

The JPEG2000 compression standard was recently developed to incorporate advances in image compression technology and better serve the digital imaging applications in the internet era. The new standard offers superior low bit-rate performance, continuous-tone and bilevel compression, protective image security, among other features.

One of the key technical differences between JPEG and JPEG2000 is the adoption of the discrete wavelet transform (DWT) [37] instead of the 8×8 block-based discrete cosine transform (DCT) [43]. The DWT, in general, offers better compression performance over the DCT.

The DWT decomposes the original image at multiple orientations and at different scales. This is similar to several models of the human visual system [10], [27]. This offers more opportunities to incorporate better models of human vision in the JPEG2000 standard. Compared to JPEG, JPEG2000 includes more visual optimization tools. The quantization matrices in the original JPEG standard take into account the threshold-versus-intensity and contrast sensitivity function of the human visual system. Besides threshold-versus-intensity and contrast sensitivity function, JPEG2000 also includes visual masking as part of the visual optimization tools [49].

Image compression algorithms have traditionally exploited the properties of the human visual system to increase the compression ratio. Watson [45] develops an image compression algorithm based on human vision, especially contrast masking. The recent JPEG2000 standard includes visual masking as part of the visual optimization tools. This extension can improve image quality over textured regions. This nonlinearity is inserted between the forward wavelet transform and the quantization module at the decoder, and a "masking compensation" module is added after the dequantization at the decoder. After the DWT, the coefficients go through a nonlinearity, as shown in the following:

$$z_i = \frac{sign(x_i)|x_i|^{\alpha}}{m_i} = \frac{sign(x_i)|x_i|^{\alpha}}{1 + a\sum_{k-near-i}|x_k|^{\beta}/|\phi_i|}, \qquad (1)$$

where x_i are the original coefficients after wavelet transformation, α, a, and β are constants that control the effects of visual masking, and $|\phi_i|$ is the size of the neighborhood. The JPEG2000 recommended values for α, β, and a are 0.7, 0.2, and $(10,000/2^{bit\_depth-1})^{\beta}$, respectively.

The denominator of (1) describes the importance of masking in a neighborhood. In the original JPEG2000 standard, the neighborhood includes only the coefficients available to the current pixel (causal neighborhood) at the decoder. However, since we have access to all the neighborhood pixels, in our computation, we use all the neighborhood pixels for the current pixel. This equation takes into account the artificial edges that commonly exist in the images, and it computes a small importance value for those cases. This helps to protect coefficients around sharp edges. In our case, the masking importance values around sharp edges will assume smaller values.

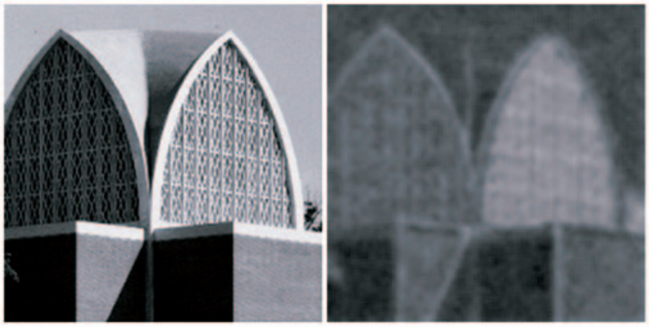


Fig. 5. The chapel image and its importance map calculated using the method inspired by the JPEG2000 standard.

The denominator of (1) can be applied to every band (except for the base band) after the discrete wavelet transformation. The masking elevation factors for each band at a specific scale describe the masking power for that band at that scale. To compute an average masking power, we take an average of the elevation factors at each band and each scale. Fig. 5 shows the chapel image and its masking importance map. Note that the windows of the chapel have larger importance values than the relatively uniform background regions.

There are two main differences between our method and that of Ramasubramanian et al. [35]. First, our method uses the DWT, while their method uses the DCT. The DWT better mimics the multiscale and multiorientation nature of the human visual system. Second, these two approaches use different masking functions. As mentioned before, our approach includes a mechanism that suppresses large coefficients due to edges in an image. This effect can be observed in the importance image in the right image of Fig. 5.

4.3 Discussion

The two algorithms described in Sections 4.1 and 4.2 share a few similarities. First, both algorithms decompose the original image with a multiscale and multiorientation decomposition scheme, which parallels the workings of the human visual system. Second, both approaches use a similar visual masking function to calculate the visual masking potential of a texture.

However, there are significant differences between them. First, the two visual masking evaluation algorithms were based on two very different algorithms. The Sarnoff VDM was designed to discriminate the visual differences between two images in an imaging system due to physical parameter changes, whereas the JPEG2000 inspired algorithm was designed for perceptually guided image compression. Second, the two algorithms use different multiscale and multiorientation decomposition schemes. While the algorithm based on JPEG2000 uses DWT, the Sarnoff VDM uses a decomposition that better fits with the psychophysical data. Third, the two algorithms use different units in the importance map. The one based on JPEG2000 uses physical luminance error. In contrast, the Sarnoff VDM uses the unit of JNDs, where one JND means 75 percent of probability of being seen as an artifact. Another important difference is that, in order to use the Sarnoff VDM, we have to generate a second image, but the JPEG2000-based algorithm only requires a single image. Lastly, the JPEG2000-based algorithm uses the DWT, and it is much faster in terms of computational speed and is more memory efficient than the Sarnoff-based algorithm.

5 Perceptually Guided Remeshing

The input to the remeshing algorithm is assumed to be a parameterized triangulated mesh and several surface signals that have accumulated on the mesh during rendering. First, our algorithm generates the composite surface signal from several surface signal sources. Our algorithm then analyzes the perceptual properties of the composite surface signal using the importance map evaluation methods described in Section 4. Third, the surface mesh is converted to a map-based representation, and the geometry remeshing process is treated as a 2D sampling process based on an importance map. Finally, a Delaunay triangulation operation is performed on these samples. These samples and their connectivity are reprojected back to 3D to form a 3D mesh.

We introduce the algorithm by showing how it can account for a single type of observable surface signal: the color pattern produced by 2D texture mapping. We then broaden the definition of the surface signal to include the effect of such things as bump mapping, spotlighting, shadow patterns, and interreflections. We also demonstrate how all of the effects included in this general definition of the surface signal can be accommodated using the same procedures developed to handle 2D texture mapping.

5.1 Map-Based Representation

Once we compute the masking importance map for the texture, we can take advantage of this information to perform geometry remeshing. In this paper, we have adapted the remeshing approach developed by Alliez et al. [3]. This method computes a set of 2D maps to represent the geometric properties of the model. The advantage of this technique is that most of the remeshing and filtering operations can be easily done in the 2D parametric domain.

To represent the geometric properties of the model, we have computed the following 2D maps: an area distortion map, a curvature map, and a regular sampling of the 2D parametric domain [3]. When combined with the previously computed visual discrimination map, we can perform perceptually based geometry remeshing.

5.2 Importance Sampling Based on Centroidal Voronoi Tessellation

A density map is computed using the maps determined previously. Ideally, high curvature areas and low visual masking texture areas require denser sampling, while low curvature and strong visual masking areas require less sampling. We have used two parameters to guide the generation of the density map. The curvature gamma adjusts the relative importance of the curvature map. The visual perceptual gamma adjusts the relative importance of the visual perceptual map.

Once the density map is computed, we need to discretize the density map to a set of samples. Alliez et al. [3] use error diffusion to generate the samples, then switch to centroidal Voronoi tessellation [2]. In this work, we take the second approach because it generates highly regular samples.

Given a region A and a density function ρ defined over this region, the mass of centroid c of A is defined by

$$c = \frac{\int_A x\rho(x)\mathrm{d}x}{\int_A \rho(x)\mathrm{d}x}. \tag{2}$$

One way to compute the weighted centroidal Voronoi tessellation is to use Lloyd's relaxation [26]. Lloyd's relaxation can be considered as a fixed point iteration. Given a density map and an initial set of n sites, it consists of the following three steps:

1. Build a Voronoi diagram of the n sites.
2. Compute the centroid of each site and move the n sites to their respective centroid.
3. Repeat steps 1 and 2 until a satisfactory solution is reached.

Efficiently computing the centroid of each site is not a trivial problem. Determining the centroid requires evaluation of (1) for each site. Inspired by the work of Hoff et al. [16], we resort to the use of computer graphics hardware to compute the centroid of each site. A fragment program is used to perform the integrations in (2) using the vector reduce operation together with summation [20]. There is one major issue with computing Voronoi diagram using graphics hardware. The number of samples in a Voronoi diagram can exceed the number of pixels that the graphics hardware supports. This is especially true for our case since there can be millions of samples for large models. To get around this issue, we compute one centroid at a time instead of computing the centroids of all sites at the same time.

5.3 Remeshing Results

The left image in Fig. 6a demonstrates the result of rendering the Igea model (refer to Fig. 7e) with a leaf texture pattern into a 2D map. The masking importance map that corresponds to this 2D map is shown in Fig. 6b. Centroidal Voronoi tessellation is then used to generate samples (shown in Fig. 6c) from Fig. 6b.

Perceptually based surface remeshing of the texture-mapped Igea model depicted in Fig. 7 begins with the generation of samples from the masking importance map shown in Fig. 6b. The result of applying Lloyd's relaxation on the map for 20 iterations is illustrated in Fig. 6c. Finally, the samples are reprojected to 3D to generate the 3D mesh.

Fig. 7 shows the remeshing of the Igea model both with and without using the visual perceptual properties of the surface signal. The two pairs (Figs. 7a and 7b and Figs. 7c and 7d) are generated with different gamma values for the curvature and perceptual components. The original mesh contains 5,000 vertices. Fig. 7a shows a uniformly remeshed model (curvature gamma is 0) with 2,000 vertices. Fig. 7b is produced with curvature gamma 0 and perceptual gamma 1.0. Fig. 7c is produced with curvature gamma 1.2. Fig. 7d is produced with the same curvature gamma as Fig. 7c and perceptual gamma 2.0.

Notice that the geometric details on the top part of the original mesh are further removed as shown in Figs. 7b and

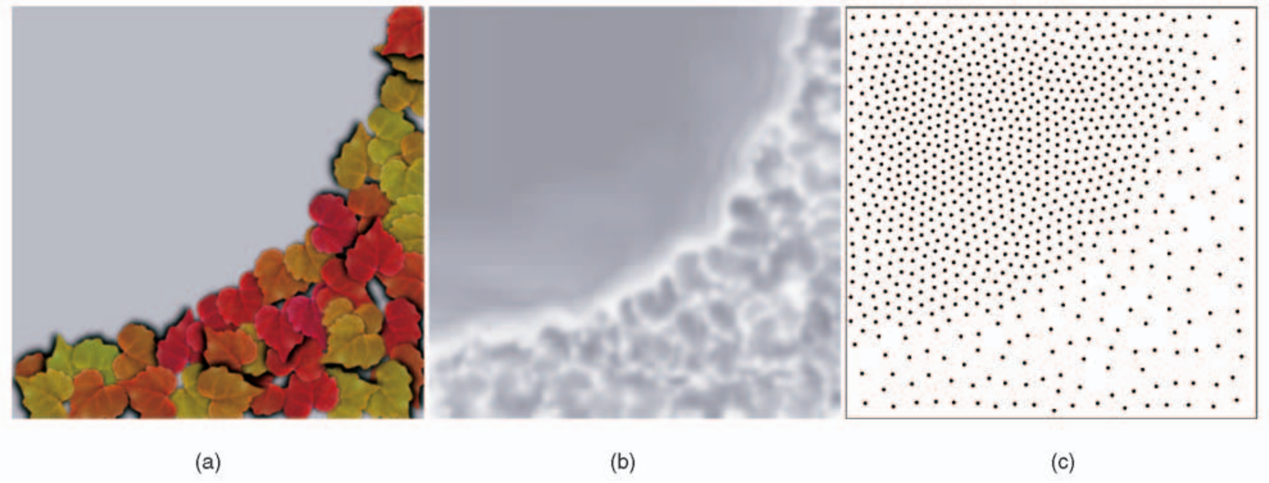


Fig. 6. (a) Is the original texture, (b) is its masking importance map, and (c) shows the samples generated using centroidal Voronoi tessellation from (b).

7d compared to Figs. 7a and 7c, respectively, since it is covered by texture. This further reduction of polygon count in textured areas will not be noticeable due to the visual masking properties of the texture. The triangles saved in the textured areas are used in other parts of the model. As can be seen in the figure, the eyes, nose, and mouth of the model have denser samples than the model without the perceptual component, thus more details are preserved in these areas.

On a Xeon 1.8 GHz with 1-G memory, it takes 0.8 second to compute the surface signal and convert the geometry into a map-based representation. The evaluation of the Sarnoff VDM takes about 4.2 seconds for an image pair of size 512×512. The centrical Voronoi tessellation using 10 Lloyd's iterations for Fig. 7 takes about 20 seconds. This is the most expensive part of the algorithm. Fortunately, very few iterations are required to generate good samples. Furthermore, generating samples using image halftoning techniques can be used in the design phase to create the initial samples.

5.4 Other Types of Surface Signals

Two-dimensional texture mapping has been used in this paper to demonstrate how the perceptual properties of the texture, such as masking, can be used to guide the remeshing of the geometry to which the texture has been applied. However, 2D texture mapping is only one example of several processes that combine to produce the final color pattern that is seen on the surface of the object. We call the variation of lightness and color that is seen by a viewer looking at the object, and that is generated by mechanisms independent of the underlying geometry, the surface signal. To achieve the most dramatic reduction in polygons the complete surface signal should be used in the remeshing process.

In this section, we enumerate the processes by which the surface signal can be altered. In each case, we demonstrate how our approach makes use of a single framework to exploit the resulting surface signal and decrease the number of polygons in the underlying geometric mesh. We note that some of the methods by which the surface signal is altered are viewpoint independent and could be taken into account once for a static background like those found in most animations and video games. In other viewpoint-dependent cases, one would need to page in different mesh representations or remesh on the fly as the observer's position was changed.

5.4.1 Viewpoint-Independent Surface Signals

Bump mapping is another means by which the surface signal can be altered without manipulating the underlying geometry. When only simple diffuse shading is used to perform the bump mapping, the result will be viewpoint independent. Fig. 8 demonstrates that the illusion of an embossed pattern on the surface due to bump mapping can have a masking effect similar to that produced by 2D texture mapping. The area beneath the embossing requires fewer polygons than the homogeneous surfaces adjacent to the embossed area.

Variations in the intensity of a light source across a surface can be another component of the surface signal. The most straightforward way for this to happen is when the light source is focused into a spotlight. This can produce a bright spot on the surface and raise the visual threshold within that pool of light. An example of this is given in Fig. 9. Here, we see that fewer polygons are required within the bright region produced by the spotlight. Alternatively, obstructions in front of a light source can produce intensity variations that have a similar impact. Masking effects are even possible, as shown in Fig. 11, when the pattern of shadows has the necessary frequency content. Here, the required number of polygons is reduced in the shadowed areas.

5.4.2 Viewpoint-Dependent Surface Signals

Evaluation of a surface reflection model is an obvious way to alter the surface signal. Implicit in the viewpoint-independent surface signals described above is a diffuse shading calculation. Here, we consider the effect of adding a strong specular term to the reflection model that is employed. The result can be a bright highlight on the surface of the object. In a manner similar to the spotlight discussed above, the number of polygons required beneath the specular highlight is reduced because the visual threshold has been elevated in

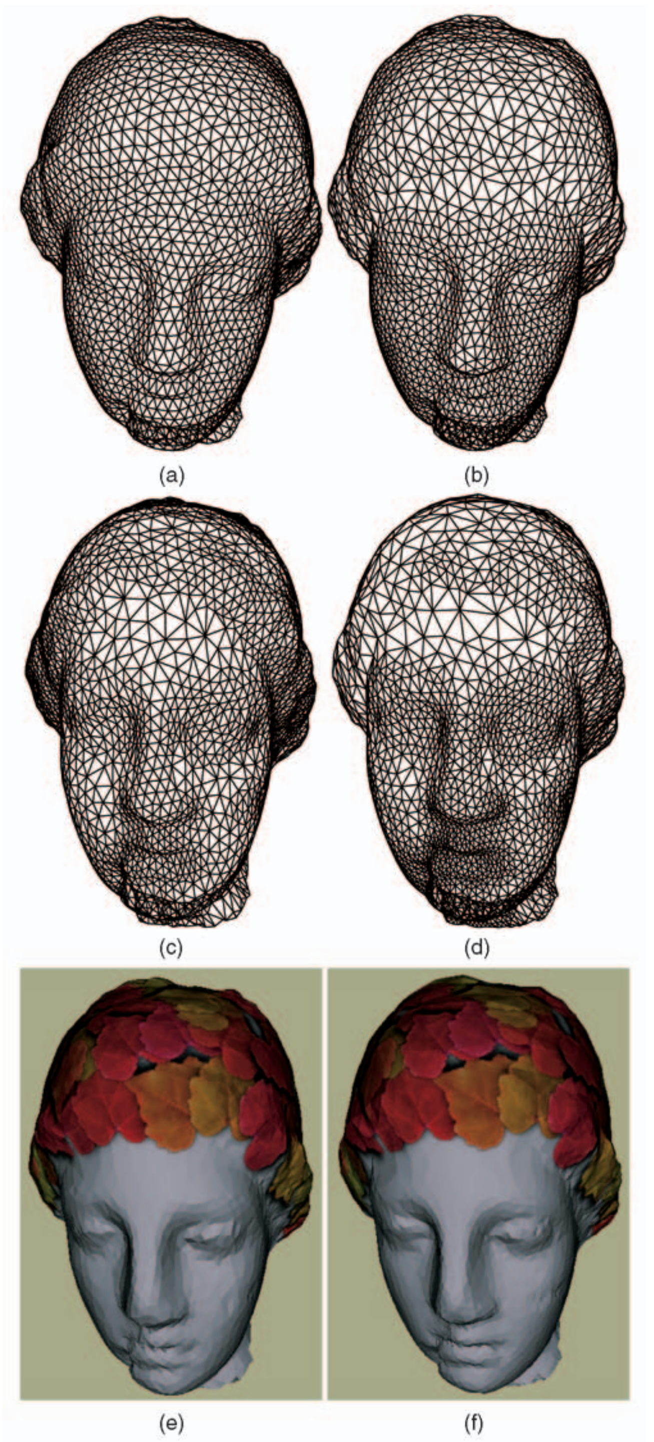


Fig. 7. (a) Is a uniformly remeshed model. (b) Is a uniformly remeshed model with perceptual component. Notice that the details on the top part of the original mesh are further removed since it is covered by texture. The same for pairs (c) and (d) but with different gamma values. (e) and (f) Are rendered images of models shown in (c) and (d), respectively. Notice that more details can be seen in the eyes, nose, and mouth area in (f).

this region. Other more complex BRDFs may produce surface signal variations that can also be exploited to reduce the number of polygons in a mesh.

When the specular reflection becomes even stronger and interreflections are calculated, the surface signal will include the reflected image of other objects in the environment. These mirror reflections will produce a pattern on the surface that can be exploited to reduce the number of polygons in the object mesh. An example of this is shown in Fig. 10. Here, the shiny teaspoon that reflects the surrounding environment requires fewer polygons than the diffuse teaspoon. This illustration was produced using an environment mapping technique to simulate the interreflections. It is interesting to note that an environment map that might not produce a masking effect as a 2D surface texture can create a surface signal that will mask the underlying polygons when it is distorted by reflection onto a surface.

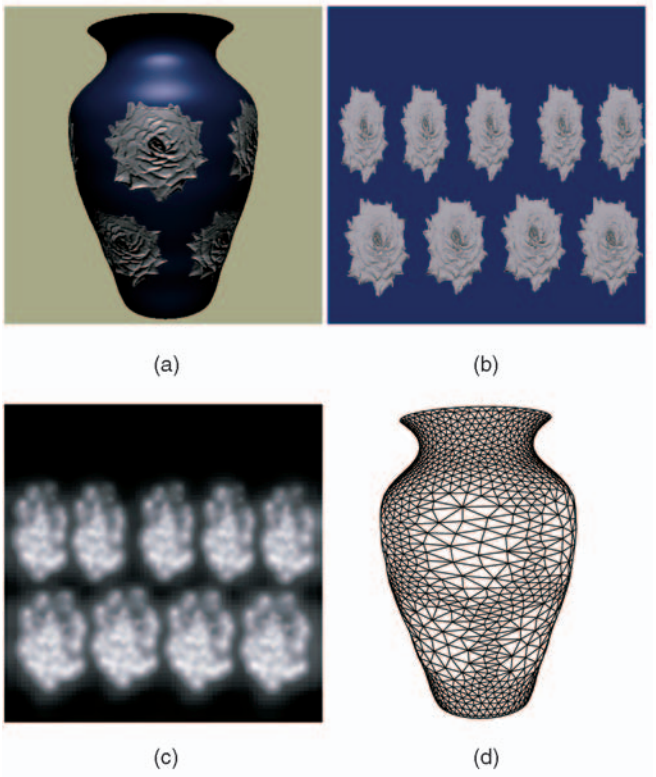


Fig. 8. Bump-mapped vase (a) created using a normal map and a vase model. The shading calculation transforms the normal map into a color pattern which is gathered into a (b) color map. The perceptual properties of the color map are then evaluated using a VDM. The brighter region in map (c) indicates stronger visual masking. This map is then used to guide the placement of vertex samples (d) in the geometry remeshing stage. The original vase has 7,171 vertices, and the remeshed vase has 2,000 vertices.

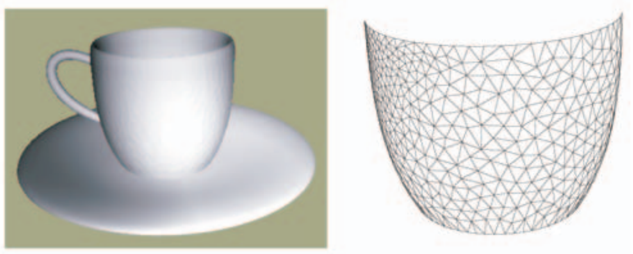


Fig. 9. Spotlighted region raises the visual threshold and decreases the number of polygons required.

5.4.3 Complex Scenes

As a final example, we present a complex scene composed of multiple objects where several different types of surface

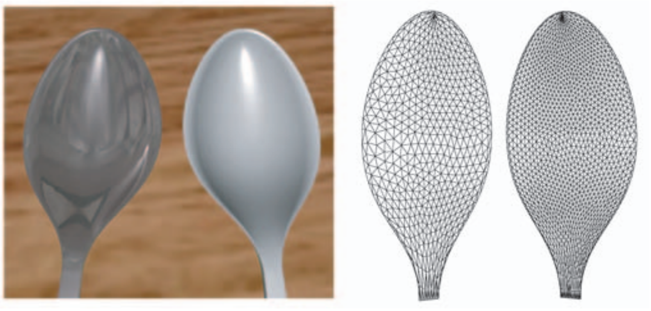


Fig. 10. Reflections produce a masking pattern and reduce the number of polygons required in the mesh. The shiny teaspoon on the left has only 1,027 vertices, and the diffuse teaspoon on the right has 2,761 vertices.

signals occur in combination. In contrast to the modest gains that can be achieved in the above examples where only a single surface signal is exploited, many opportunities present themselves to recover polygons in a complicated environment where texture mapping is used extensively, surface reflection properties vary widely, and there are several different light sources. A background scene for a computer video game is an example of an environment where there are numerous surface signals, where the objects do not move, and where the viewpoint remains relatively constant. In this case, the polygons that are collected from the background objects can be used to create a more detailed moving foreground object that is the center of the viewer's attention.

In Fig. 11, we demonstrate how the algorithm presented in this paper can be used to simplify the meshes in a complicated scene that has multiple surface signals. All of the objects in this picture were decimated using our perceptually based remeshing algorithm. Examples of the mesh reduction achieved for the cup and the teapot are shown in this figure.

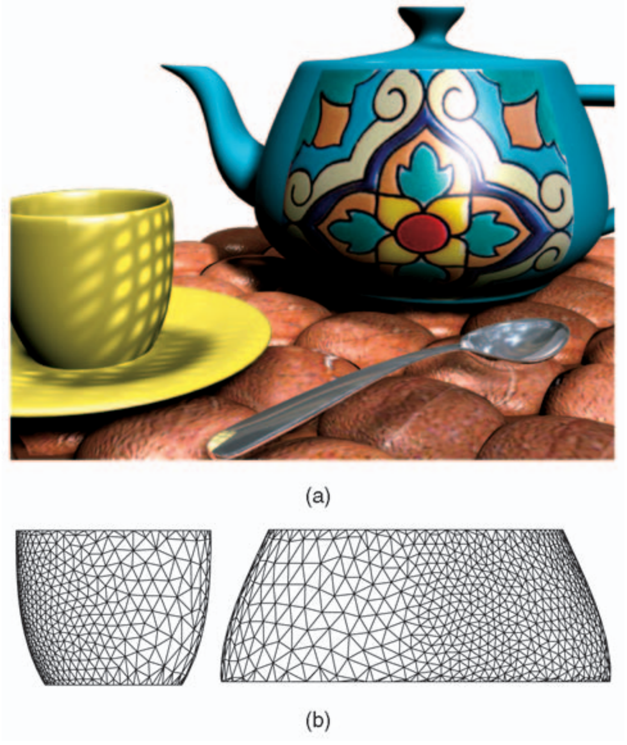


Fig. 11. All surfaces in (a) are modeled using quadric and cubic surfaces. (b) Shows example of the remeshing achieved for the cup and the teapot (rotated to show the texture boundary).

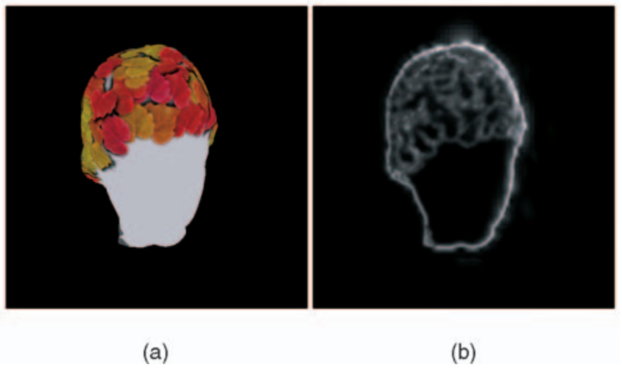


Fig. 12. (a) A slice of the surface light field and (b) its masking importance map.

6 Perceptually Guided Simplification

In this section, we describe a perceptually guided mesh simplification algorithm. This algorithm begins by taking snapshots of a geometric object in its desired lighting environment. These snapshots serve as a coarse representation of the surface light field of the geometric object. Then, the analysis phase of the algorithm computes the masking potential importance map for each snapshot using the visual masking importance evaluation algorithms described in Section 4. Third, an integration process is used to determine an importance value for each vertex of the original mesh. Finally, this per-vertex importance map is integrated into a geometry-based mesh simplification algorithm (QSlim [13]) by taking into account the surface geometry and the masking importance map.

6.1 Light Field Masking Potential

The appearance of an object is fully captured by the light field of the object. The light field represents the radiance value as a function of position and direction. Several light field representation schemes have been proposed in the computer graphics literature [15], [23], [48]. A sphere of cameras surrounding an object also offers a representation of the light field of an object. This representation has been used in an image-driven simplification approach [25].

In this paper, we choose to use the sphere of cameras' representation for the sake of simplicity. We place a sphere of cameras at the position of a small rhombicuboctahedron, similar to the image-driven simplification [25]. The object is rendered from these camera positions with desired lighting, surface materials, and so on. Fig. 12a shows one slice of the surface light field of the Igea model under uniform lighting. This set of images is then used to represent the light field of the object. Since our goal is not to fully reconstruct the radiance values for the object (as is the case for image-based rendering), this rough approximation will suffice for our application.

Once we have a representation of the light field, the masking potential of the light field can be evaluated. The masking potential of each slice of the light field is determined using the masking potential evaluation procedure described in Section 4. Fig. 12 shows a slice of the surface light field and its masking importance map, where brighter areas indicate stronger masking.

6.2 Masking Importance Map Evaluation

Given the masking importance maps of the surface light field, we describe a method that computes an importance value for each vertex of the mesh. Depending on the reflectance properties of the surface, the light field around one point on the surface can be view dependent. Therefore, this importance value can be different for different viewing directions. In this paper, we choose to compute the average masking potential of the surface light field at the vertex of the mesh. The average masking importance I of one point p on the surface can be formulated using the following equation:

$$I(\mathbf{p}) = \int_{\Omega} \rho(\mathbf{v}) F_{p,\mathbf{v}} \mathbf{d}v, \tag{3}$$

where $\rho(\mathbf{v})$ is the weight for the masking at viewing direction $\mathbf{v}$, $F_{p,\mathbf{v}}$ is the masking importance at point p and viewing direction $\mathbf{v}$. In our implementation, ρ is a cosine of the angle between the surface normal and the viewing direction. Cameras pointing more directly to the vertex are weighted more than other cameras.

Direct computation of this integration is a nontrivial problem. To make the evaluation of (3) practical, we have taken the following two steps to simplify it. First, instead of evaluating an importance value at every point on the surface, we compute an importance value for each vertex of the original mesh. Second, we precompute the masking potential of the surface light field from a fixed number of viewing positions, as described in the previous section. To compute the importance value for a vertex v of the mesh, a set of rays are shot from vertex v to each camera position p. The intersection of a ray and the image plane of the camera can be used to index into the 2D importance map and get a value. This value is weighted by the cosine of the angle between the vertex normal and the ray. Algorithm 1 describes the pseudocode for our importance map evaluation. G represents the geometric model, and C is the set of cameras. $\mathbf{n}$ is an array of normals (per-vertex normal), and s is an array that stores the masking importance value for each vertex of the mesh. This procedure is similar to projective texturing. The 2D masking importance map associated with each camera position is projected onto the mesh. Cameras looking more directly at a vertex are awarded more weight than cameras looking at the vertex at skew angles.

Algorithm 1: Masking importance evaluation

```
for each vertex $v \in G$ do
  $s[v] \leftarrow 0$
  $w \leftarrow 0$
  for each camera $c \in C$ do
    shoot a ray $\mathbf{r}$ from vertex $v$ to camera $c$
    find the intersection of $\mathbf{r}$ and the image plane of the
    camera and fetch a precomputed importance value $m$
    $t \leftarrow \mathbf{n}[v] \cdot \mathbf{r}$
    $s[v] \leftarrow s[v] + t * m$
    $w \leftarrow w + t$
  end for
  $s[v] \leftarrow s[v]/w$
end for
```

6.3 Perceptually Guided Mesh Simplification

Once a masking importance map is computed over the geometric mesh, it can be integrated into a geometry-based mesh simplification algorithm. If the visual masking map indicates there is strong visual masking in certain areas of the mesh, the mesh in those areas can be simplified more aggressively. This is because the error introduced by more aggressive simplification can be made less visible due to masking caused by the surface signal. The masking-aware algorithm can be implemented by weighting the geometric error metric with masking importance.

In our implementation, we choose the quadratics-based simplification algorithm by Garland and Heckbert [13] as the base algorithm due to its time and memory efficiency and good approximations. Their method associates each vertex v with a set of planes $\mathbf{S}$ incident at the vertex. A quadratic Q that represents the distance from the vertex v to a plane $\mathbf{n}.\mathbf{v} + d = 0$ is given by

$$Q = (\mathbf{A}, \mathbf{b}, c) = (\mathbf{n}\mathbf{n}^{\mathbf{T}}, d\mathbf{n}, d^2).$$

The quadratic associated with vertex v is the sum of the quadrics for the plane set S and the resulting quadratic after contracting an edge (v_1, v_2) to v_3 is just the sum of quadratics for v_1 and v_2.

Similar to other methods [21], [50], we have developed a weighting scheme to adjust the quadratics associated with each vertex of the mesh. The weight W for a vertex v is defined as

$$W(v) = \frac{1.0}{1.0 + I(v)^{\alpha}}. \tag{4}$$

$I(v)$ is the masking importance at vertex v. α can be used to adjust the relative importance of the masking importance map. The weight is 1.0 if the masking importance value is zero, and it becomes smaller as the importance values increases. We have found that an α value of 2.0 gives good results.

6.4 Simplification Results

Fig. 13 shows the simplification of a teapot model with a flower pattern. Fig. 13a shows an image rendered using the original model. Fig. 13b is the masking importance map computed using our algorithm. Figs. 13c and 13d are rendered with simplified meshes using QSlim and our modified QSlim algorithm, respectively. Both models have 8,000 triangles. The images are rendered using flat shading in order to show its polygons. Figs. 13e and 13f are rendered with wireframe superimposed on their textured renderings. Notice that the polygons underneath the flower pattern are bigger in Fig. 13d without causing more visible artifacts than Fig. 13c. To maintain similar appearance for a simplified model, we have used the vertex attribute transfer tool in *Maya* which transfers the texture coordinates of the original model to its simplified model for the examples we show in this section.

Fig. 14 shows the simplification of the Igea model. Fig. 14a shows an image rendered using the original

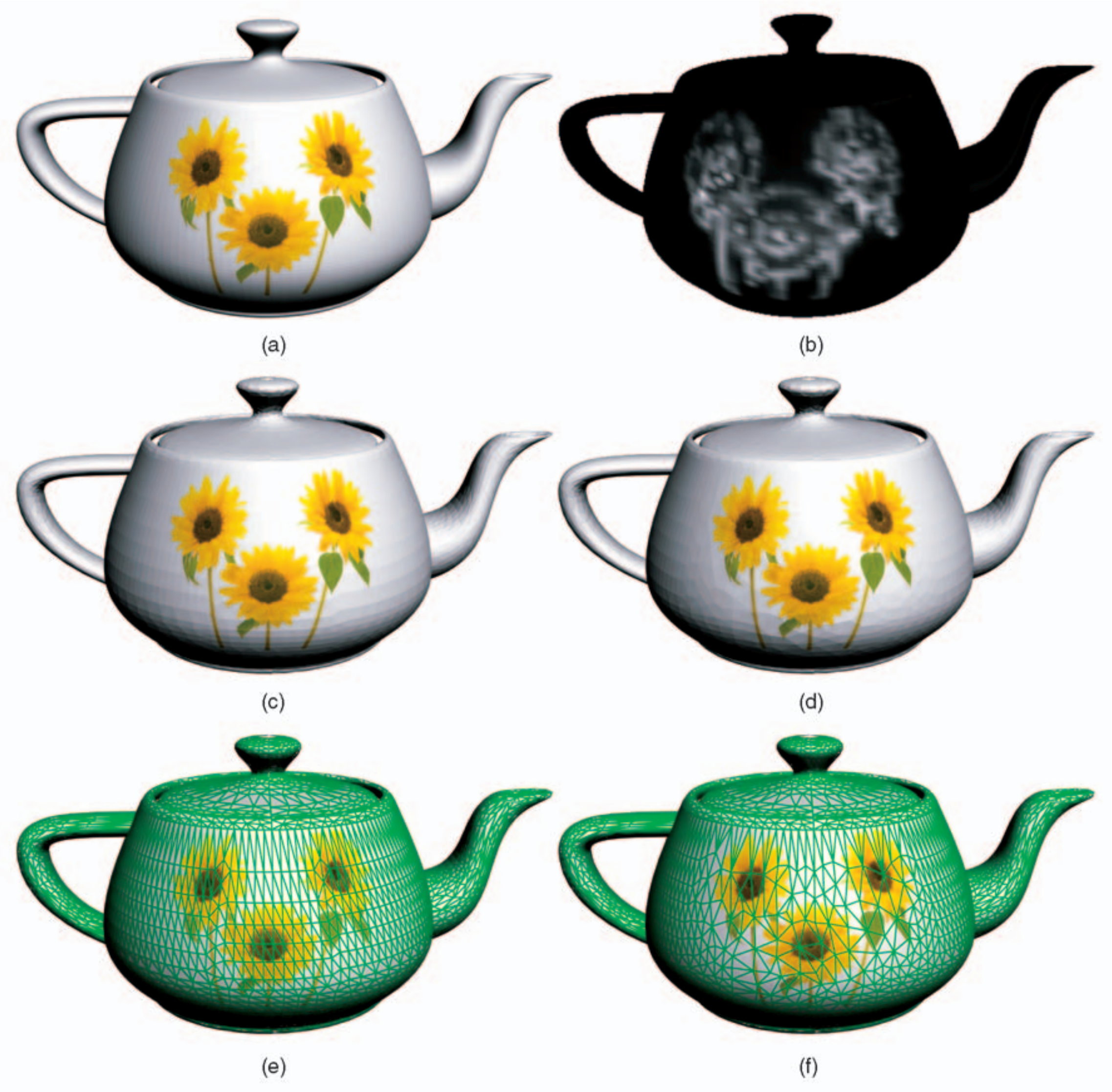


Fig. 13. (a) A rendering of the original model. (b) Masking importance map. (c) Rendered using a simplified mesh with QSlim. (d) Rendered using a simplified mesh with QSlim taking into account the masking potential of the surface light field. Both models have the same number of vertices. Their respective wireframe renderings are shown in (e) and (f).

model. Fig. 14b is the masking importance map computed by our algorithm. Figs. 14c and 14d are rendered using simplified meshes with the original QSlim method and our modified QSlim method, respectively. Both models have 2,000 triangles. Figs. 14e and 14f are rendered with wireframe superimposed on their textured renderings. Notice that the polygon size in the textured areas is bigger for our perceptually guided mesh simplification algorithm.

Fig. 15a shows a simple scene with two bunnies. They are covered with a ceramic texture and a fabric pattern, respectively. Their masking importance map is shown in Fig. 15b. Notice that the importance values for the left bunny are smaller (darker) than the right bunny. Given 6,000 triangles as the budget for this scene, each bunny contains roughly 3,000 faces for QSlim. However, for the modified QSlim algorithm, the left bunny contains 3,456 triangles and the right bunny contains 2,544 triangles in the simplified scene. Figs. 15e and 15f are rendered with wireframe superimposed on their textured renderings. Note the difference in polygon size for the two bunnies in Fig. 15f. The right bunny has approximately 1,000 triangles less than the left bunny. Our algorithm automatically allocates the number of polygons for each mesh in a scene by taking into account the masking potential of its surface signal.

Table 1 shows the timing information of our algorithm as executed on a Xeon machine with 1 G of memory. Our algorithm has a preprocessing phase besides the simplification phase. The preprocessing phase includes evaluating the masking potential of its surface light field (denoted as VDM in the table) and then assigning the masking potential to the vertices of a mesh (denoted as Importance Map in the table).

7 Discussion

Our perceptually guided remeshing and simplification algorithms share a similar framework: both use a precomputed importance map to guide the remeshing/simplification process. However, the remeshing algorithm computes an importance map in the parametric domain, whereas the simplification algorithm computes importance

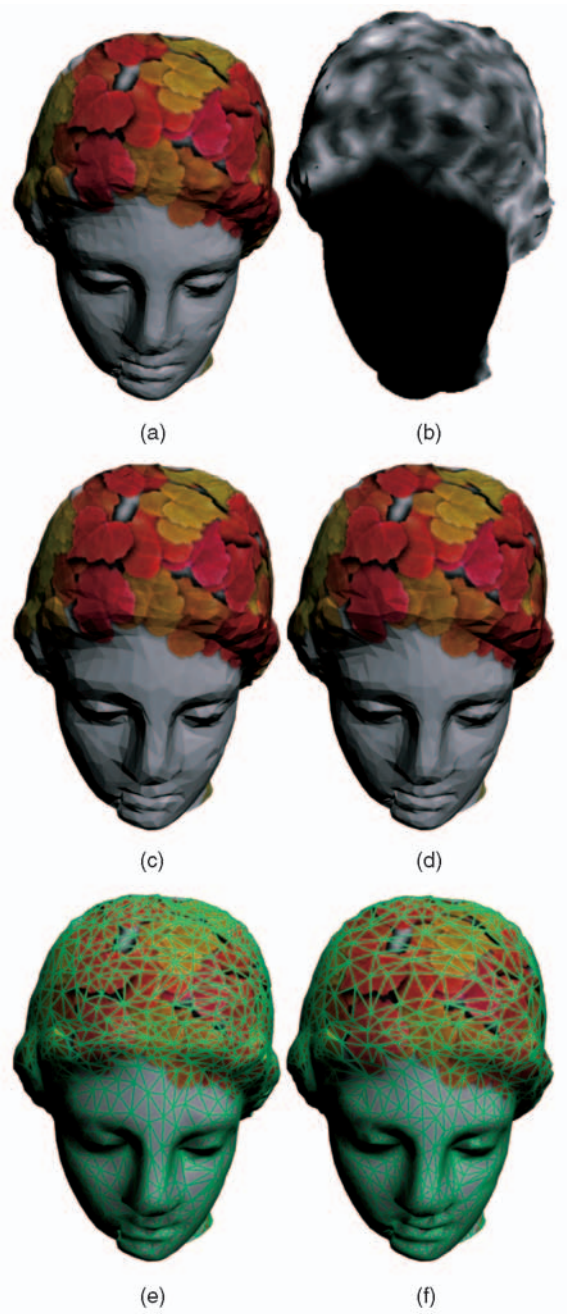


Fig. 14. (a) Is a rendering of the original model. (b) Is the masking importance map. (c) Is rendered using a QSlim simplified mesh. (d) Is rendered using a mesh simplified with QSlim while taking into account the masking potential of the surface light field. Both models have the same number of vertices. Their respective wireframe renderings are shown in (e) and (f).

values over the vertices of the geometric model in object space. In addition, the remeshing algorithm has to deal with parametric distortion, while mesh simplification algorithms do not have this disadvantage.

The biggest difference of our work from previous work in perceptually driven mesh simplification [28], [47] is that we have taken visual masking into account. Visual masking is a very strong phenomenon and can be used to further decrease the sampling rate of the geometry in certain regions (refer to Fig. 1) without causing visible artifacts. It is also much more expensive to compute than the other aspects of the visual system such as the contrast sensitivity function and the threshold-versus-intensity function. State-of-the-art visual masking algorithms [10], [12], [27] require multiscale and multiorientation decomposition of the images. Considering the amount of computation required to evaluate a full perceptual metric including visual masking, it is unrealistic to evaluate this metric for each resampling or edge collapse operation for a moderate size geometry with currently available hardware.

To overcome this limitation, we resort to computing an importance map that suggests the visual masking potential of the surface signal in the texture space for our surface remeshing algorithm. This importance map can then be used to guide the remeshing process. In the case of our mesh simplification algorithm, we take slices of the surface light field of the object, compute the masking potential, and map the computed masking potential to the vertices of the mesh. The importance values on the vertices are then used to guide the mesh simplification algorithm, similar to some previous importance-based mesh simplification algorithms [19], [21], [30], [50]. Thus, we avoid the evaluation of masking potential for each edge collapse for the mesh simplification algorithm.

Our work also differs significantly from previous work in mesh simplification that preserves appearance properties such as colors, positions, and normals of geometric models [8], [9], [14]. Appearance-preserving simplification algorithms compute new vertex positions and texture coordinates so as to minimize both geometric error and texture stretch. Our approach focuses on the reduction of the polygon count by taking advantage of the perceptual properties of the surface signals. Both approaches take surface signals into account, but with different goals. We do not claim that our methods perform better than the aforementioned approaches [8], [9], [14]. However, we do believe that appearance-preserving simplification algorithms can be further improved if the perceptual properties of the surface signals are taken into account.

It is also worth mentioning that the remeshing and simplification algorithms presented in this paper do not address the silhouette edge issue of simplified geometric models. In addition, our work considers the textures and general lighting level of a static scene, and it is not our intention to take into account dynamic lighting.

8 Summary

In this paper, we have presented a perceptually driven surface remeshing algorithm. Our remeshing algorithm automatically distributes samples uniformly over a polygon mesh by taking the perceptual properties of the surface signal into account during the remeshing process. Due to the properties of the human visual system, especially visual masking, the artifacts in the final rendered mesh are invisible to the human observer. We have also shown that there are many opportunities, besides simple 2D texture mapping, to exploit the masking properties of the surface signal and redistribute the polygons available to render a scene. Among the additional mechanisms that contribute to the surface signal are bump mapping, spot lighting, shadow patterns, surface reflectance, and interreflections.

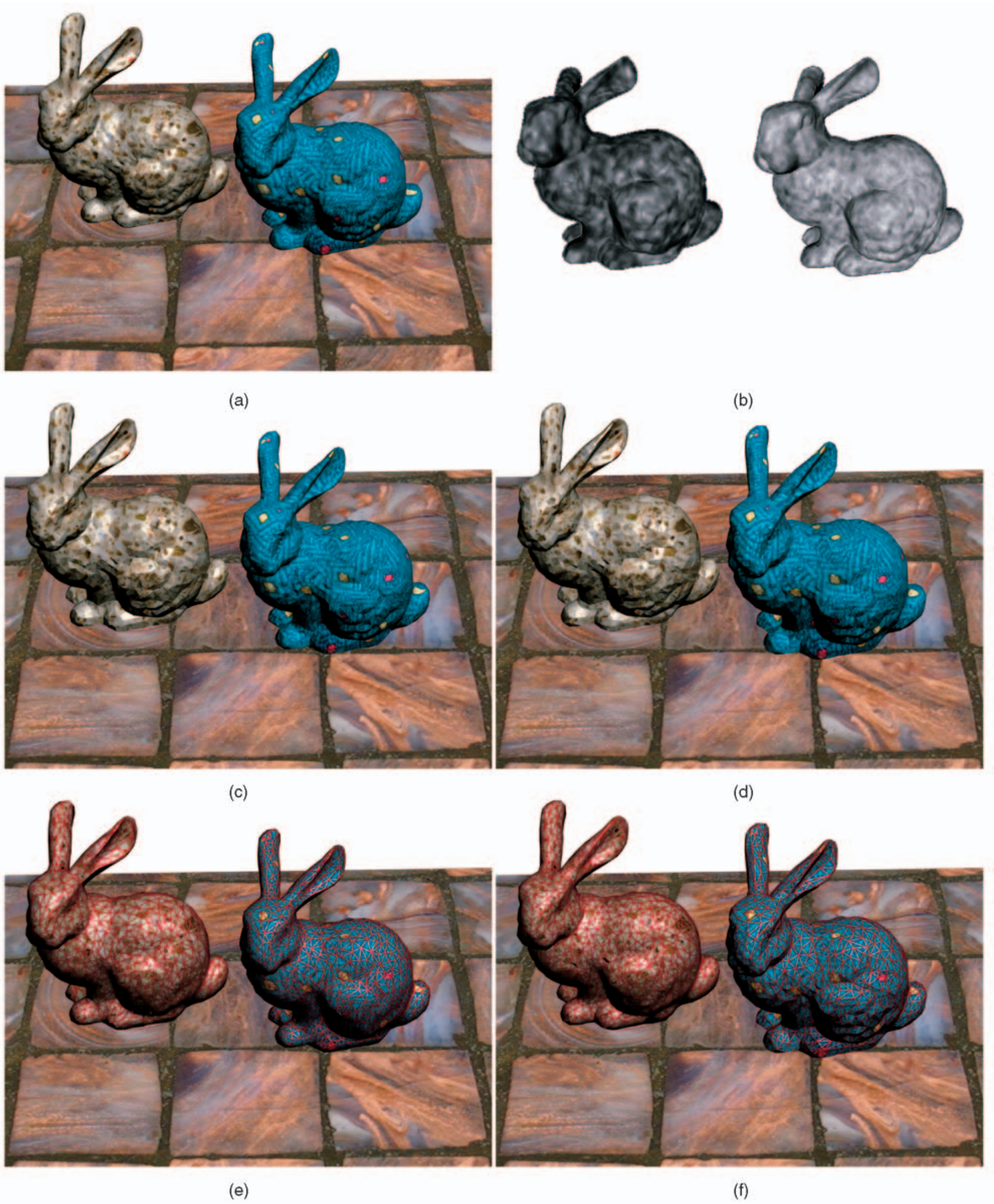
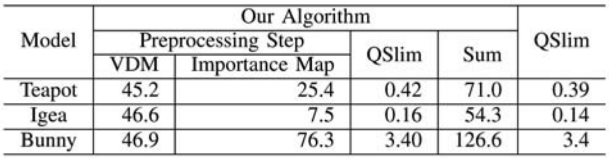


Fig. 15. (a) Is the original scene with two bunnies. (b) Shows the masking importance map. (c) Is rendered using the simplified meshes with QSlim. (d) Is rendered using the simplified mesh with our modified QSlim. Their corresponding wireframe renderings are shown in (e) and (f).

We have also demonstrated a novel mesh simplification method that takes into account the visual masking potential of the surface light field of a geometric object. Due to the masking caused by the surface light field of an object, certain regions of the mesh can be simplified more aggressively than the other regions without causing visual artifacts. Our method is general enough to account for lighting, surface materials, etc. This work is a continuation of current research in mesh simplification that takes into account factors other than geometry such as visibility, mesh saliency, and semantic features.

While this paper has primarily used isolated individual objects to demonstrate the effectiveness of perceptual-based polygon reduction algorithms, there is an even greater

TABLE 1
Performance of Our Perceptually Guided Mesh Simplification Algorithm (in Seconds)



Model	Our Algorithm				QSlim
	Preprocessing Step		QSlim	Sum	
	VDM	Importance Map			
Teapot	45.2	25.4	0.42	71.0	0.39
Igea	46.6	7.5	0.16	54.3	0.14
Bunny	46.9	76.3	3.40	126.6	3.4

potential for efficiency gains when these objects are placed in context. In a video game or an animated film where many of the objects and much of the lighting in the scene remains static, a large number of the polygons allocated for the background objects can be recovered and used to render principal characters or objects in the foreground. This can reduce rendering times and improve the overall quality of the final animated sequence.

In the future, we would like to study the properties of the artifact signal due to coarse tessellations [32]. We believe such study can provide insight that can further improve our perceptually guided surface remeshing and mesh simplification algorithms.

Acknowledgments

The authors would like to thank all anonymous reviewers for their helpful comments and suggestions, which have significantly improved this paper. They also want to thank Michael Garland for making his QSlim software available, and Cyberware and Stanford Graphics Lab for making the Igea and Bunny models available. The assistance of Jon Konieczny in running timing tests is also gratefully acknowledged. Funding was provided by the National Science Foundation Grant CCR-0242757. This research was performed at the University of Minnesota Digital Technology Center.

References

[1] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, and M. Desbrun, "Anisotropic Polygonal Remeshing," *Proc. ACM SIGGRAPH '03,* pp. 485-493, 2003.

[2] P. Alliez, É.C. de Verdiére, O. Devillers, and M. Isenburg, "Isotropic Surface Remeshing," *Proc. Shape Modeling Int'l (SMI '03),* pp. 49-58, 2003.

[3] P. Alliez, M. Meyer, and M. Desbrun, "Interactive Geometry Remeshing," *Proc. ACM SIGGRAPH '02,* pp. 347-354, 2002.

[4] L. Balmelli, G. Taubin, and F. Bernardini, "Space-Optimized Texture Maps," *Proc. Eurographics '02,* pp. 411-420, 2002.

[5] M.R. Bolin and G.W. Meyer, "A Perceptually Based Adaptive Sampling Algorithm," *Proc. ACM SIGGRAPH '98,* pp. 299-309, 1998.

[6] N.A. Carr and J.C. Hart, "Painting Detail," *Proc. ACM SIGGRAPH '04,* pp. 845-852, 2004.

[7] I. Cheng and P. Boulanger, "Adaptive Online Transmission of 3D TexMesh Using Scale-Space Analysis," *Proc. Second Int'l Symp. 3D Data Processing, Visualization and Transmission (3DPVT '04),* pp. 688-695, 2004.

[8] P. Cignoni, C. Montani, C. Rocchini, R. Scopigno, and M. Tarini, "Preserving Attribute Values on Simplified Meshes by Resampling Detail Textures," *The Visual Computer,* vol. 15, no. 10, pp. 519-539, 1999.

[9] J. Cohen, M. Olano, and D. Manocha, "Appearance-Preserving Simplification," *Proc. ACM SIGGRAPH '98,* pp. 115-122, 1998.

[10] S. Daly, "The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity," *Digital Images and Human Vision,* MIT Press, pp. 179-206, 1993.

[11] J.A. Ferwerda, S.N. Pattanaik, P. Shirley, and D.P. Greenberg, "A Model of Visual Adaptation for Realistic Image Synthesis," *Proc. ACM SIGGRAPH '96,* pp. 249-258, 1996.

[12] J.A. Ferwerda, P. Shirley, S.N. Pattanaik, and D.P. Greenberg, "A Model of Visual Masking for Computer Graphics," *Proc. ACM SIGGRAPH '97,* pp. 143-152, 1997.

[13] M. Garland and P.S. Heckbert, "Surface Simplification Using Quadric Error Metrics," *Proc. ACM SIGGRAPH '97,* pp. 209-216, 1997.

[14] M. Garland and P.S. Heckbert, "Simplifying Surfaces with Color and Texture Using Quadric Error Metrics," *Proc. IEEE Visualization (VIS '98),* pp. 263-269, 1998.

[15] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, "The Lumigraph," *Proc. ACM SIGGRAPH '96,* pp. 43-54, 1996.

[16] K.E. Hoff, J. Keyser, M. Lin, D. Manocha, and T. Culver, "Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware," *Proc. ACM SIGGRAPH '99,* pp. 277-286, 1999.

[17] S. Howlett, J. Hamill, and C. O'Sullivan, "An Experimental Approach to Predicting Saliency for Simplified Polygonal Models," *Proc. First Symp. Applied Perception in Graphics and Visualization (APGV '04),* pp. 57-64, 2004.

[18] S. Howlett, J. Hamill, and C. O'Sullivan, "Predicting and Evaluating Saliency for Simplified Polygonal Models," *ACM Trans. Applied Perception,* vol. 2, no. 3, pp. 286-308, 2007.

[19] Y. Kho and M. Garland, "User-Guided Simplification," *Proc. Symp. Interactive 3D Graphics (SI3D '03),* pp. 123-126, 2003.

[20] J. Krueger and R. Westermann, "Linear Algebra Operators for GPU Implementation of Numerical Algorithms," *Proc. ACM SIGGRAPH '03,* pp. 908-916, 2003.

[21] C.H. Lee, A. Varshney, and D.W. Jacobs, "Mesh Saliency," *Proc. ACM SIGGRAPH '05,* pp. 659-666, 2005.

[22] G.E. Legge and J.M. Foley, "Contrast Masking in Human Vision," *J. Optical Soc. America,* vol. 7, pp. 1458-1471, 1980.

[23] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proc. ACM SIGGRAPH '96,* pp. 31-42, 1996.

[24] P. Lindstrom, "Model Simplification Using Image and Geometry-Based Metrics," PhD thesis, Georgia Inst. Technology, 2000.

[25] P. Lindstrom and G. Turk, "Image-Driven Simplification," *ACM Trans. Graphics,* vol. 19, no. 3, pp. 204-241, 2000.

[26] S.P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Information Theory,* vol. IT-28, no. 2, pp. 129-137, 1982.

[27] J. Lubin, "A Visual Discrimination Model for Imaging System Design and Evaluation," *Vision Models for Target Detection and Recognition.* World Scientific, pp. 245-283, 1995.

[28] D.P. Luebke and B. Hallen, "Perceptually-Driven Simplification for Interactive Rendering," *Proc. 12th Eurographics Workshop Rendering Techniques (EGRW '01),* pp. 223-234, 2001.

[29] C. O'Sullivan, S. Howlett, Y. Morvan, R. McDonnell, and K. O'Conor, "Perceptually Adaptive Graphics," *Proc. Eurographics '04,* State of the Art Reports, pp. 141-164, 2004.

[30] E. Pojar and D. Schmalstieg, "User-Controlled Creation of Multiresolution Meshes," *Proc. Symp. Interactive 3D Graphics (SI3D '03),* pp. 127-130, 2003.

[31] L. Qu and G.W. Meyer, "Perceptually Driven Interactive Geometry Remeshing," *Proc. Symp. Interactive 3D Graphics and Games (I3D '06),* pp. 199-206, 2006.

[32] L. Qu and G.W. Meyer, "Faceting Artifact Analysis for Computer Graphics," *Proc. 15th Pacific Conf. Computer Graphics and Applications (Pacific Graphics '07),* pp. 419-422, 2007.

[33] L. Qu, X. Yuan, M.X. Nguyen, G.W. Meyer, B. Chen, and J.E. Windsheimer, "Perceptually Guided Rendering of Textured Point-Based Models," *Proc. IEEE/Eurographics Symp. Point-Based Graphics (PBG '06),* pp. 95-102, 2006.

[34] G. Ramanarayanan, J. Ferwerda, B. Walter, and K. Bala, "Visual Equivalence: Towards a New Standard for Image Fidelity," *ACM Trans. Graphics,* vol. 26, no. 3, p. 76, 2007.

[35] M. Ramasubramanian, S.N. Pattanaik, and D.P. Greenberg, "A Perceptually Based Physical Error Metric for Realistic Image Synthesis," *Proc. ACM SIGGRAPH '99,* pp. 73-82, 1999.

[36] P.V. Sander, S.J. Gortler, J. Snyder, and H. Hoppe, "Signal-Specialized Parametrization," *Proc. 13th Eurographics Workshop Rendering Techniques (EGRW '02),* pp. 87-98, 2002.

[37] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 Still Image Compression Standard," *IEEE Signal Processing Magazine,* vol. 18, pp. 36-58, 2001.

[38] W.A. Stokes, J.A. Ferwerda, B. Walter, and D.P. Greenberg, "Perceptual Illumination Components: A New Approach to Efficient, High Quality Global Illumination Rendering," *Proc. ACM SIGGRAPH '04,* pp. 742-749, 2004.

[39] V. Surazhsky and C. Gotsman, "Explicit Surface Remeshing," *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Processing (SGP '03),* pp. 20-30, 2003.

[40] G. Tewari, J. Snyder, P. Sander, S. Gortler, and H. Hoppe, "Signal-Specialized Parameterization for Piecewise Linear Reconstruction," *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Processing (SGP '04),* pp. 55-64, 2004.

[41] G. Turk, "Re-Tiling Polygonal Surfaces," *Proc. ACM SIGGRAPH '92,* pp. 55-64, 1992.

[42] V. Volevich, K. Myszkowski, A. Khodulev, and E.A. Kopylov, "Using the Visual Differences Predictor to Improve Performance of Progressive Global Illumination Computation," *ACM Trans. Graphics,* vol. 19, no. 2, pp. 122-161, 2000.

[43] G.K. Wallace, "The JPEG Still Picture Compression Standard," *Comm. ACM,* vol. 34, no. 4, pp. 30-44, 1991.

[44] B. Walter, S.N. Pattanaik, and D.P. Greenberg, "Using Perceptual Texture Masking for Efficient Image Synthesis," *Computer Graphics Forum,* vol. 21, no. 3, pp. 393-400, 2002.

[45] A.B. Watson, "The Cortex Transform: Rapid Computation of Simulated Neural Images," *Computer Vision, Graphics, and Image Processing,* vol. 39, no. 3, pp. 311-327, 1987.

[46] B. Watson, A. Friedman, and A. McGaffey, "Measuring and Predicting Visual Fidelity," *Proc. ACM SIGGRAPH '01,* pp. 213-220, 2001.

[47] N. Williams, D. Luebke, J.D. Cohen, M. Kelley, and B. Schubert, "Perceptually Guided Simplification of Lit, Textured Meshes," *Proc. Symp. Interactive 3D Graphics (SI3D '03),* pp. 113-121, 2003.

[48] D.N. Wood, D.I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D.H. Salesin, and W. Stuetzle, "Surface Light Fields for 3D Photography," *Proc. ACM SIGGRAPH '00,* pp. 287-296, 2000.

[49] W. Zeng, S. Daly, and S. Lei, "An Overview of the Visual Optimization Tools in JPEG 2000," *Signal Processing: Image Comm.,* vol. 17, pp. 85-104, 2002.

[50] E. Zhang and G. Turk, "Visibility-Guided Simplification," *Proc. IEEE Visualization (VIS '02),* pp. 267-274, 2002.

Lijun Qu received the bachelor's degree in electrical engineering from Beijing Institute Technology, Beijing, the master's degree from the Graduate School of Chinese Academy of Sciences, Beijing, and the PhD degree from the University of Minnesota. He is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis. He is a member of graphics product group at Advanced Micro Devices. His research interests include 3D computer graphics in general with emphasis on the perceptual aspects of geometry simplification, rendering, and surface texturing.

Gary W. Meyer received the bachelor's degree from the University of Michigan, the master's degree from Stanford University, and the PhD degree from Cornell University. He is an associate professor in the Department of Computer Science and Engineering, University of Minnesota, Minneapolis. He has also been a member of the Computer Science faculty at the University of Oregon and a member of Technical Staff at Bell Telephone Laboratories. His research interests include the synthesis of color and appearance in computer graphic pictures, perceptual issues related to synthetic image generation, and color reproduction and color selection for the human-computer interface.