Eurographics Symposium on Point-Based Graphics (2006)
M. Botsch, B. Chen (Editors)

# Perceptually Guided Rendering of Textured Point-based Models

Lijun Qu[†] Xiaoru Yuan[†] Minh X. Nguyen[†] Gary W. Meyer[†] Baoquan Chen[†] Jered E. Windsheimer[‡]

Department of Computer Science & Engineering, University of Minnesota at Twin Cities, MN, USA
Digital Technology Center, University of Minnesota at Twin Cities, MN, USA

**Abstract**

*In this paper, we present a textured point-based rendering scheme that takes into account the masking properties of the human visual system. In our system high quality textures are mapped to point-based models. Given one texture, an importance map is first computed using the visual masking tool included in the JPEG2000 standard. This importance map indicates the masking potential of the texture. During runtime, point-based models are simplified and rendered based on this computed importance. In our point simplification method, called Simplification by Random Numbers (SRN), each point in the model is pre-assigned a random value. During rendering, the pre-assigned value is compared with the preferred local point density (derived from importance) to determine whether this point will be rendered. Our method can achieve coherent simplification for point models.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation

## 1. Introduction

Points have been used as alternative modeling and rendering primitives for over a decade and have recently received increasing attention from the computer graphics community [LW85]. Points have been shown to be flexible and efficient in representing highly detailed features on complex objects [GD98,PG01,PKKG03,PZvBG00,RL00]. Working directly on point-based geometries greatly simplifies content creation, surface reconstruction and rendering, as no connectivity and topological constraints exist.

One recent addition to the point based rendering literature is high resolution texture mapping for point based models [NYN*06]. This makes it possible to map a high resolution texture onto a low resolution point based model.

Perceptually guided rendering has been a very important research topic in computer graphics, especially for expensive global rendering algorithms [BM98, RPG99, WPG02, SFWG04]. Properties of the human visual system have also been used to control the level of detail of polygonal meshes [Red01] and to compute a simplified polygonal mesh [LH01, WLC*03, QM06a, QM06b]. However, we are not aware of any work in the point based modeling and rendering literature that takes into account the properties of the human visual system.

In this paper, we develop a method that brings together research in visual perception in computer graphics and point based graphics. More specifically, we propose a perceptually guided level of detail system for point based models. The basic idea of our perceptually guided level of detail system is that fewer point samples can be sent to the graphics card and rendered in certain textured areas. Because of the masking properties of the visual system, the shading artifacts (due to reduced point samples) that would otherwise be visible, can be masked or hidden by the high resolution textures on the point surface. This can lead to higher rendering speeds.

Our system first computes a masking importance map for each texture based on the visual optimization tools in the recent JPEG2000 standard. This masking importance map indicates the pixel-wise masking potential of the texture. This importance map is then used to modulate a level of detail system for point based models and fewer samples are rendered in textured areas than for other areas.

Our contributions in this paper are twofold. First, we propose a method that computes the masking potential of a

[†] {lijun,xyuan,mnguyen,meyer,baoquan}@cs.umn.edu
[‡] Now at Cryptic Studios, CA, USA

texture based on the visual optimization tools in the recent JPEG2000 standard. Second, we develop a simple but effective level of detail system for point based models.

The remainder of the paper is organized as follows. We first briefly review some related work in Section 2. Then we present a introduction to visual discrimination metrics in Section 3. In Section 4, we describe a point based level of detail system guided by the Sarnoff visual discrimination metric. We then present a novel method that evaluates the masking potential of a texture based on the JPEG2000 standard in Section 5. Section 6 describes a simple but effective way to control the density of the points based on the importance map. Finally, we present some results in Section 7 and some discussion and future work in Section 8.

## 2. Related Work

Visual perception has been exploited in computer graphics to accelerate global illumination algorithms. By taking advantage of the properties of the human visual system, computational resources can be intelligently allocated where they are most needed. Bolin and Meyer [BM98] suggest a perceptually adaptive sampling algorithm that samples the image plane based on the image content in the rendered images. Other researchers [RPG99, WPG02] develop novel methods that compute the masking capabilities of texture maps, which are then used to guide the rendering process. Stokes et al. [SFWG04] propose perceptual illumination components, where a metric is developed that predicts the perceptual importance of the computationally expensive indirect illumination components with respect to image quality.

Luebke and Hallen [LH01] present a perceptually driven mesh simplification algorithm that controls the simplification using psychophysical models of visual perception. Each edge collapse operation is mapped to a worst contrast sinusoidal grating. The simplification decision is made by taking into account the contrast sensitivity (CSF) of the human visual system. They later extended their work to the simplification of lit, textured meshes [WLC*03]. However, visual masking is not included, possibly due to the prohibitive cost required by the visual masking evaluation, which in general requires expensive multi-scale, multi-orientation decomposition of the images. Qu and Meyer propose a remeshing algorithm [QM06a] and a mesh simplification algorithm [QM06b] that account for the masking properties of the human visual system. Higher level of visual perception such as visual saliency has also been incorporated into a mesh simplification algorithm [LVJ05].

On the point-based rendering side, many algorithms focus on efficiency and speed [GD98] using hardware acceleration [RL00, RPZ02]. Emphasis has also been placed on high quality rendering. The surface splatting technique [PZvBG00, ZPvBG01, RPZ02] and differential points [KV01, KV03] are proposed for high-quality surface rendering. Alexa et al. [ABCO*01] control the fidelity of the representation by dynamically adjusting the density of the points. Zwicker et al. [ZPvBG01] introduce the Elliptical Weighted Average (EWA) filter to increase the rendering quality for point-based rendering. Qsplat [RL00] progressively displays points using a multiresolution hierarchy based on bounding spheres. Compact data representation is applied with storage and computation efficiency for large data sets. Stamminger and Drettakis [SD01] dynamically adjust the point sampling rate for rendering for procedural and complex geometries. In later work multi-resolution rendering mixing points and polygons has been proposed by Chen and Nguyen [CN01] and Cohen et al. [CAZ01]. In the above methods, determination of the data level is based on the projected size on the screen. Properties of the human visual system are not considered.

## 3. Visual Discrimination Metrics (VDM)

The human visual system is often modeled as a series of processes that limit the resolution of the visual system. Even though there are new discoveries about the human visual system, researchers in general agree that information processing of the human visual system includes three stages: threshold-vs-intensity function, contrast sensitivity function, and visual masking. The threshold-vs-intensity function describes the non-linearity 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 in the presence of a background object. Several visual discrimination metrics [Dal93, Lub95, FSPG97] have been developed that incorporate the above mentioned properties of the visual system.

In this paper, we are primarily interested in the visual masking properties of the human visual system. Visual masking refers to the phenomenon where a background signal hides or masks the visibility of a foreground signal. This is a very strong phenomenon, and has been used in the imaging industry for a long time. Artifacts introduced into the original image can be considered as the foreground object, and the original image is considered as the background object. For example, visual masking has been used in image compression algorithms [Wat87]. Quantization can be performed more aggressively in areas of strong frequency content without introducing visual artifacts. Quantization noise caused by the aggressive quantization can be masked by the image content.

In a visual masking experiment, 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. Figure 1 shows the effect of visual masking. The abscissa represents the contrast of the masker, and the ordinate represents the contrast of the sig-

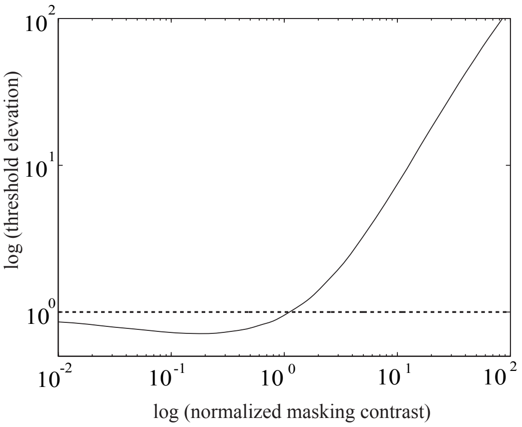


**Figure 1:** *Visual masking function of the visual system, adapted from Ramasubramanian et al. [RPG99].*

nal. As can be observed in the diagram, when the contrast of the masker is low, there is no or little masking. Masking becomes apparent as the contrast of the masker increases.

## 4. Lighting and Point Based Rendering

As an initial demonstration of applying visual perception to point based rendering systems, we converted the user-controlled error metric of the hybrid of Points and Polygons (POP) system [CN01] to the JND (Just Noticeable Differences) of the Sarnoff VDM [Lub95]. The POP system is continuous, view-dependent, and runs at interactive rates. POP utilizes a hybrid rendering method, drawing the data as both points and polygons. Points are faster to render than polygons, so when polygons are small a point can be a good approximation to the polygon.

POP uses a hierarchical data structure, with the polygonal mesh as the leaves of a tree, and the intermediate nodes represent bounding spheres that are rendered as points. Each intermediate node is placed and sized to completely encompass its child nodes. In this way an intermediate node represents the space around its children. The threshold of the POP system serves as part of its error metric. When the projected screen size of a intermediate node is smaller than the threshold, the node is drawn as a point and its children are ignored. If a node is larger than the threshold it is not drawn and all its children are examined. In this way the threshold acts as a measure of the tradeoff between visual quality and rendering speed.

To integrate the Sarnoff VDM with the POP model, we wanted to find some relationship between JND and viewing parameters that could be evaluated in real-time. To achieve this we chose to sample the space of possible light source positions and pre-calculate the JND at each sample. Rather than forcing the error metric to do a full evaluation of the VDM for each frame, the error metric can use the current light source location as an index into a table of pre-calculated lighting situations. Each of these sampled illumination conditions has already been analyzed by the VDM, and the relevant information such as mean JND can be retrieved and used.

For the task of integrating the Sarnoff VDM with the POP model, the Stanford Bunny was used as the object, drawn on a flat black background and lit by a spotlight. The spatial position of the spotlight varied over 11 positions on the X and Y axes and three distances from the model. Images were generated at 10 different threshold values for each position of the spotlight. The Sarnoff VDM was then used to find the mean JND between the lowest threshold image and the remaining nine images. Figure 2 shows the variation in the mean JND as the light moves around a stationary model drawn at the center of the image with a POP threshold of 2.4. When the light is directly in front of the model, making the entire model visible and well-lit, the mean JND is at its highest (Figure 3). As the light moves away from the model, the mean JND drops, indicating that a human viewer is less likely to notice errors (Hence, larger POP threshold values, Figure 4).

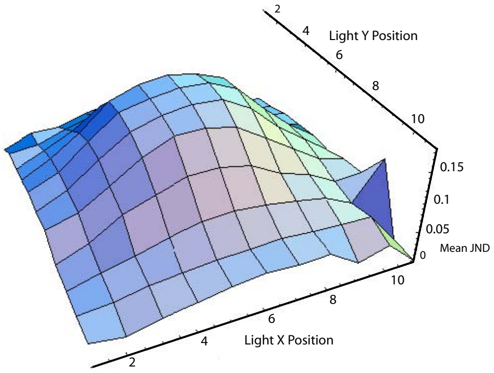
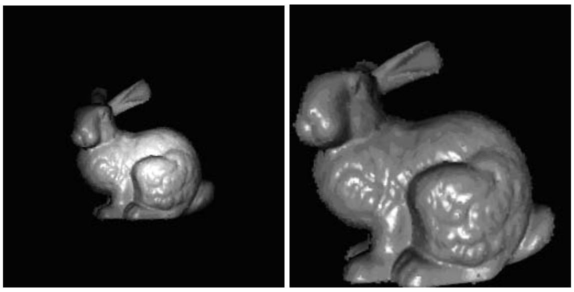


**Figure 2:** *Graph of variation in mean JND as the light moves in the X-Y plane, with the model drawn using a POP threshold of 2.4.*

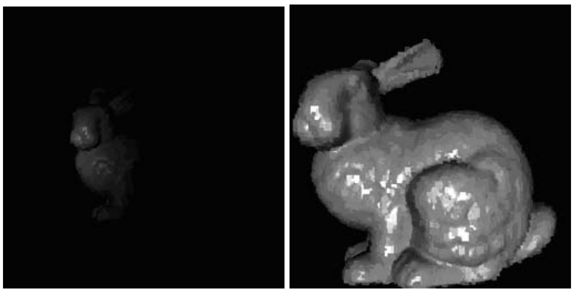
## 5. Masking Importance Evaluation

Given a texture, the masking importance evaluation computes an importance map that indicates the pixel-wise masking capabilities of the texture. Several approaches have been proposed to evaluate the masking potential of a texture in the computer graphics literature. Ramasubramanian et al. [RPG99] develop a method that evaluates the masking potential of a texture. Walter et al. [WPG02] suggest an approach that determines the masking potential of a texture using JPEG operators. More recently, Qu and Meyer [QM06a] developed a method based on the Sarnoff visual discrimination metric [Lub95]. In this paper, we suggest another masking evaluation method based on the visual optimization tools available in the recent JPEG2000 standard [ZDL02].



(a) (b)

**Figure 3:** *(a) View of the POP system with a centered light. A POP threshold of 1.931 was selected. (b) Scaled and re-lit copy of the actual points and polygons rendered for this scenario.*



(a) (b)

**Figure 4:** *(a) View of the POP system with an off-center light. A POP threshold of 2.517 was selected. (b) Scaled and relit copy of the actual points and polygons rendered for this scenario.*

### 5.1. Visual Optimization Tools in JPEG2000

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) [SCE01] instead of the $8\times8$ block based discrete cosine transform (DCT) [Wal91]. The DWT in general offers better compression performance over the DCT.

The discrete wavelet transform decomposes the original image at multiple orientations and at different scales. This is similar to several models of the human visual system [Dal93, Lub95]. 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-vs-intensity and contrast sensitivity function of the human visual system. Besides threshold-vs-intensity and contrast sensitivity function, JPEG2000 also includes visual masking as part of the visual optimization tools [ZDL02].

### 5.2. Masking Importance Map

Image compression algorithms have traditionally exploited the properties of the human visual system to increase the compression ratio of the images. Watson [Wat87] developed 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 areas like texture 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 de-quantization at the decoder. After the discrete wavelet transform, the coefficients go through a nonlinearity, as shown in Equation 1 [ISO].

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

where $x_i$ are the original coefficients after wavelet transformation, $\alpha$, $\beta$, and $a$ are constants that control the effects of visual masking, and $|\phi_i|$ is the size of the neighborhood. The JPEG2000 recommended values for $\alpha$, $\beta$, and $a$ are 0.7, 0.2, and $(10000/2^{bit\_depth-1})^{\beta}$, respectively.

The denominator of Equation 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.

The denominator of equation 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 describes 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. Figure 5 shows the masking importance map for the Barbara image. Notice that the hair region has larger importance values than for the relatively uniform background regions.

There are two main differences between our method and that of Ramasubramanian et al. [RPG99]. First, our method uses the discrete wavelet transform while their method uses the discrete cosine transform. Discrete wavelet transform better mimics the multi-scale, multi-orientation nature of the human visual system. Second, these two approaches use different masking functions. As mentioned before, our approach includes mechanism that suppresses large coefficients due to edges in an image. This effect can be observed in the importance image in Figure 5.

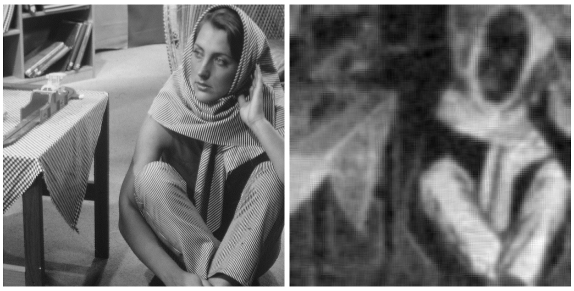


**Figure 5:** *The Barbara image and its importance map.*

## 6. Perceptually Guided Point Simplification and Rendering

We change the density of the points based on the previously computed visual masking importance map. The density control can be considered as simplification of the original point models. Simplification of point-sampled models has been introduced by Alexa et al. [ABCO*01], Linsen [Lin01] and Pauly et al. [PGK02]. Many point based rendering approaches use hierarchical data structures as a multiresolution representation. For such structures, the simplification relies on the traversal of the hierarchical data. To reduce the CPU load of data traversal, Dachsbacher et al. [DVS03] introduced a data structure called the sequential point tree that allows adaptive rendering of point clouds completely on the graphics processor.

In this paper, we introduce an extremely simple but effective method for point-based model simplification based on local density control called Simplification by Random Numbers (SRN). For a point-based model, point samples $p_1, p_2, \cdots, p_n \in R^3$ are taken in 3D space and irregularly distributed on the object's surface. We set the initial point sample density as unit 1. The computed importance map assigns a new density map $I(x), x \in R^3$ for the surface represented by the point set. During pre-processing, we first assign each point $p_i$ of the the entire surface point list with a random number $r_i, r_i \in [0,1]$. During runtime, for each point $p_i$, we compare the pre-assigned random value of $r_i$ with its corresponding density value $I(x)$. Points are rendered according following density test:

$$\begin{array}{ll} r_i < I(x) & \text{pass} \\ r_i \geq I(x) & \text{not pass} \end{array} \tag{2}$$

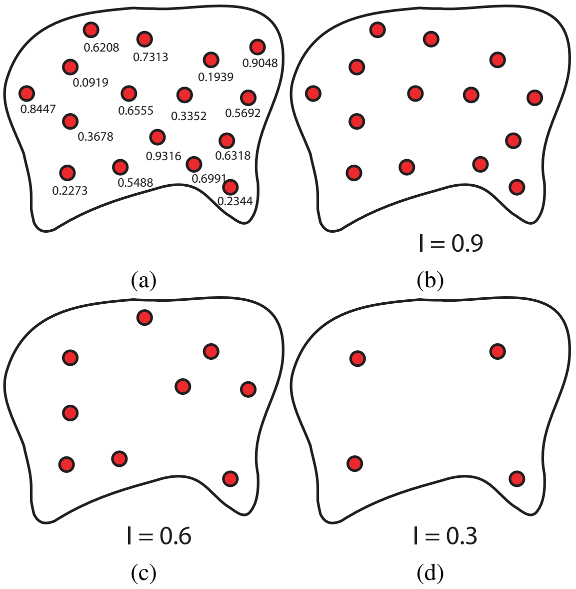


**Figure 6:** *Simplification by Random Numbers (SRN). (a) Every point is assigned a random number ranging from 0.0 to 1.0. (b)-(d) Points chosen to render based on local density values $I = 0.9$, 0.6 and 0.3 respectively.*

Only points that pass the test (with $r_i < I(x)$) are kept for rendering. In this way, the point density is controlled by the density values. Notice that as the point density decreases, the radii of each point needs to be increased accordingly to avoid holes. Our scheme is illustrated in Figure 6. Every point in the figure has been assigned a random number ranging from 0.0 to 1.0. In Figure 6 (b)-(d), a local density value of $I = 0.9$, 0.6 and 0.3 is applied respectively. Note that the point set is simplified according to the value of the density. A similar method has been used to control stroke density for non-photorealistic rendering [NXYC03, YC04]. For the point-based model simplification, however, the size of each point is adjusted in inverse proportion to the local density. The size of points in a lower density region are increased to ensure no holes. Using this method, the point-based model simplification becomes a simple number comparison operation.

We render the perceptually simplified point-based models with texture using a recently developed technique called *Texture Splatting* [NYN*06]. In our method, the texture coordinates of each point are treated as color when the point is splatted. Texture coordinates between points are thus automatically interpolated through point splatting. In the following rendering pass, colors are looked up in the texture(s) based on the interpolated texture coordinates for screen pixels. In this way, high quality texture mapping for point-based models is achieved. The importance computed using perceptual considerations can reduce the number of points to be rendered while keeping the amount of visual error small.

## 7. Results

We have implemented our perceptually guided point based rendering system on a Dell Precision 530 workstation with dual Intel Xeon 1.80G Hz CPU, 1GB RAM, 4×AGP motherboard and a 256MB GeForce 7800 GTX graphics card.

Our un-optimized implementation show clear performance speedup with our perceptually guided simplification. Table 1 tabulates the timings of rendering three different

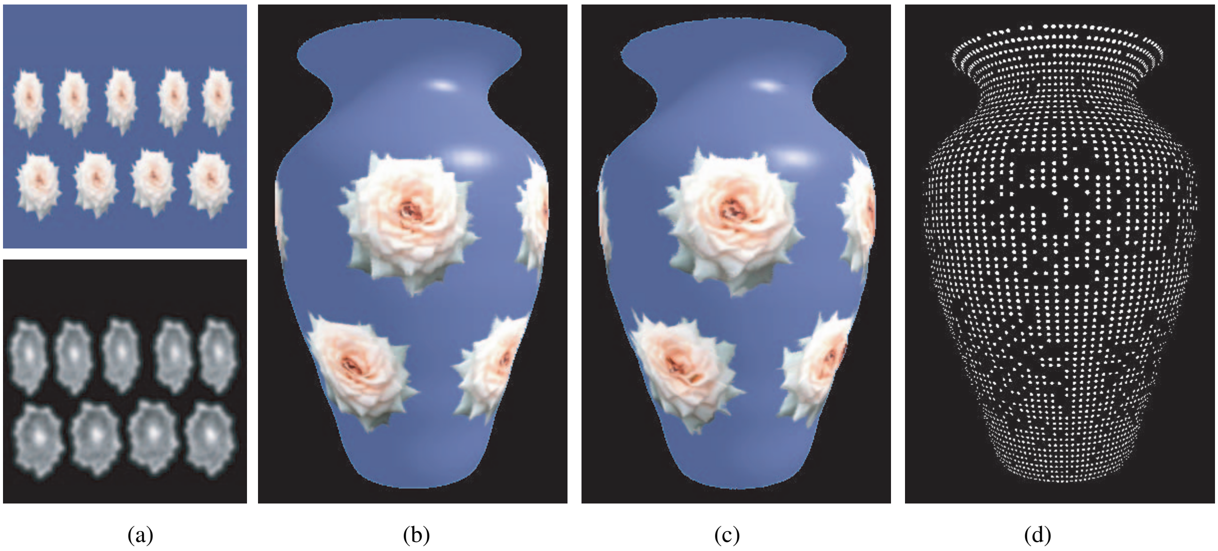


**Figure 7:** *Image (a) shows a flower pattern and its masking importance map. Image (b) and (c) are the renderings of a vase point model with texture mapping before and after simplification. Image (d) shows the point samples after simplification.*

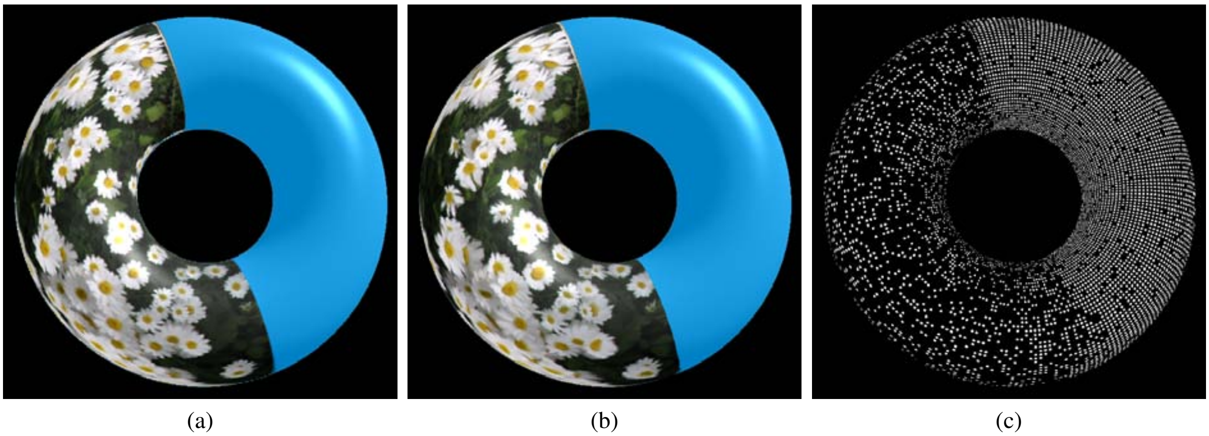


**Figure 8:** *A texture mapped torus before (a) and after simplification (b), and its point samples (c).*

| | Original | | Simplified | |
|---|---|---|---|---|
| Model | Points # | fps | Points # | fps |
| vase | 7,171 | 50.4 | 5,816 | 58.2 |
| torus | 22,321 | 22.7 | 14,426 | 33.3 |
| gargoyle | 200,004 | 2.7 | 91,437 | 6.1 |

**Table 1:** *Performance of our perceptually guided textured point-based rendering.*

point-based models. Depending on the texture content, our algorithm reduces the number of points to be rendered, thus improving the frame rate.

Figure 7 shows a flower pattern mapped onto a vase model. Figure 7(a) shows a flower pattern and its masking importance map. The brighter regions indicate stronger masking potential. Figure 7(b) and (c) show the rendered vase images using the approach by [NYN*06] before and after simplification. Figure 7(d) shows the simplified point cloud. Notice that the points are sparser underneath the flower pattern. The original model has 7,171 points, and the simplified vase model has 5,816 points.

Figure 8(a) shows a torus where half of it is textured with a flower pattern and the other half remains un-textured. 8(b) and (c) show the rendered simplified point model and its corresponding point samples. Notice that the textured area has fewer point samples than the un-textured area. The original torus model has 22,321 points, and the simplified torus has 14,426 points.

Figure 9(a) shows a scene with two gargoyle objects textured with different marble textures. The rendered point model simplified based on the perceptual properties of the marble textures is shown in Figure 9(b). Figure 9(c) illustrates the point samples of the simplified gargoyles. Due to different perceptual properties of the marble textures, the

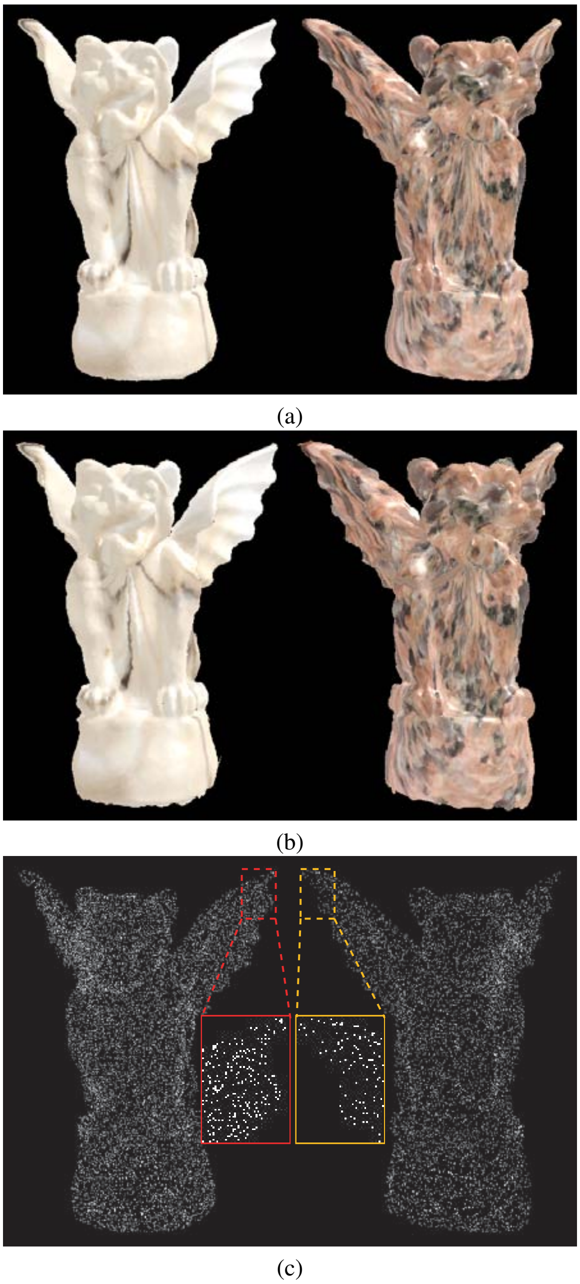


**Figure 9:** *Two gargoyles mapped with different textures before (a) and after simplification (b). Their point models are shown in image (c).*

two gargoyle objects are simplified differently. Notice that the right gargoyle contains less points than the left gargoyle. The total number of points for the two gargoyle objects is 200,004. The total number of points after simplification is 91,437, of which the left gargoyle has 54,216 points and the right gargoyle has 37,221 points. The point density difference of left and right gargoyle is illustrated in the zoom-in windows of Figure 9(c).

## 8. Conclusion and Future Work

In this paper, we have developed a method that extends a point based rendering system to include visual masking of the human visual system. For textured point models, the point density in texture regions can be reduced without generating noticeable visual artifacts in the final rendered images due to the masking properties of the visual system.

Our experiments show that the JPEG2000 based masking importance evaluation algorithm works well for rendering textured point-based models. However, a more thorough analysis of the algorithm is necessary. We plan to perform user studies to validate our approach. We also plan to compare our masking evaluation method to other methods proposed in the computer graphics literature. Other extensions include applying our masking importance map to other types of models. The density control of our point-based rendering algorithm requires the original model with regular point density to avoid holes in the final rendered image. Better density control algorithm is helpful for our point based simplification.

**Acknowledgment:** We would like to thank anonymous reviewers for their helpful comments and suggestions, and Amit Shesh for proofreading the paper. Funding supports include NSF CCR-0242757, NSF ACI-0238486 (CAREER), NSF EIA-0324864 (ITR), and NSF DMS-0528492. We thank Miao Jin for providing parameterization for the gargoyle model, and Stanford for providing the bunny model.

## References

[ABCO*01] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Point set surfaces. In *IEEE Visualization* (2001), pp. 21–28.

[BM98] BOLIN M. R., MEYER G. W.: A perceptually based adaptive sampling algorithm. In *SIGGRAPH '98* (1998), pp. 299–309.

[CAZ01] COHEN J. D., ALIAGA D. G., ZHANG W.: Hybrid simplification: combining multi-resolution polygon and point rendering. In *IEEE Visualization* (2001), pp. 37–44.

[CN01] CHEN B., NGUYEN M. X.: POP: a hybrid point and polygon rendering system for large data. In *IEEE Visualization* (2001), pp. 45–52.

[Dal93] DALY S.: The visible differences predictor: an algorithm for the assessment of image fidelity. In *Digital images and human vision*. MIT Press, 1993, pp. 179–206.

[DVS03] DACHSBACHER C., VOGELGSANG C., STAMMINGER M.: Sequential point trees. *ACM Trans. Graph. 22*, 3 (2003), 657–662.

[FSPG97] FERWERDA J. A., SHIRLEY P., PATTANAIK S. N., GREENBERG D. P.: A model of visual masking for computer graphics. In *SIGGRAPH '97* (1997), pp. 143–152.

[GD98] GROSSMAN J., DALLY W.: Point sample rendering. In *Proceedings of the 9th Eurographics Workshop on Rendering Techniques* (1998), pp. 181–192.

[ISO] ISO/IEC 15444-2:2004: *Information technology - JPEG 2000 image coding system: Extensions*.

[KV01] KALAIAH A., VARSHNEY A.: Differential point rendering. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques* (2001), pp. 139–150.

[KV03] KALAIAH A., VARSHNEY A.: Modeling and rendering of points with local geometry. *IEEE Trans. Visual. Comp. Graph. 9*, 1 (2003), 30–42.

[LH01] LUEBKE D. P., HALLEN B.: Perceptually-driven simplification for interactive rendering. In *Proceedings of Eurographics Workshop on Rendering Techniques* (2001), pp. 223–234.

[Lin01] LINSEN L.: *Point Cloud Representation*. Technical report, Faculty of Computer Science, University of Karlsruhe, 2001.

[Lub95] LUBIN J.: A visual discrimination model for imaging system design and evaluation. In *Vision Models for Target Detection and Recognition*. World Scientific, 1995, pp. 245–283.

[LVJ05] LEE C. H., VARSHNEY A., JACOBS D. W.: Mesh saliency. *ACM Trans. Graph. 24*, 3 (2005), 659–666.

[LW85] LEVOY M., WHITTED T.: *The Use of Points as Display Primitives*. Tech. Rep. TR 85-022, The University of North Carolina at Chapel Hill, Department of Computer Science, 1985.

[NXYC03] NGUYEN M. X., XU H., YUAN X., CHEN B.: INSPIRE: an interactive image assisted non-photorealistic rendering system. In *Pacific Graphics* (2003), pp. 472–477.

[NYN*06] NGUYEN M. X., YUAN X., NGUYEN T., CHEN J., CHEN B.: Texture splatting. In *Submitted to Pacific Graphics 2006* (2006).

[PG01] PAULY M., GROSS M.: Spectral processing of point-sampled geometry. In *SIGGRAPH '01* (2001), pp. 379–386.

[PGK02] PAULY M., GROSS M., KOBBELT L. P.: Efficient simplification of point-sampled surfaces. In *IEEE Visualization* (2002), pp. 163–170.

[PKKG03] PAULY M., KEISER R., KOBBELT L. P., GROSS M.: Shape modeling with point-sampled geometry. *ACM Trans. Graph. 22*, 3 (2003), 641–650.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: surface elements as rendering primitives. In *SIGGRAPH '00* (2000), pp. 335–342.

[QM06a] QU L., MEYER G. W.: Perceptually driven interactive geometry remeshing. In *ACM I3D'06* (2006), pp. 199–206.

[QM06b] QU L., MEYER G. W.: *Visual Masking Guided Simplification*. Tech. Rep. 2006/16, Digital Technology Center, University of Minnesota, 2006.

[Red01] REDDY M.: Perceptually optimized 3d graphics. *IEEE Comput. Graph. Appl. 21*, 5 (2001), 68–75.

[RL00] RUSINKIEWICZ S., LEVOY M.: Qsplat: a multiresolution point rendering system for large meshes. In *SIGGRAPH '00* (2000), pp. 343–352.

[RPG99] RAMASUBRAMANIAN M., PATTANAIK S. N., GREENBERG D. P.: A perceptually based physical error metric for realistic image synthesis. In *SIGGRAPH '99* (1999), pp. 73–82.

[RPZ02] REN L., PFISTER H., ZWICKER M.: Object space EWA surface splatting: A hardware accelerated approach to high quality point rendering. In *Computer Graphics Forum* (2002), vol. 21, pp. 461–470.

[SCE01] SKODRAS A., CHRISTOPOULOS C., EBRAHIMI T.: The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine 18* (Sep 2001), 36–58.

[SD01] STAMMINGER M., DRETTAKIS G.: Interactive sampling and rendering for complex and procedural geometry. In *Proceedings of the Eurographics Workshop on Rendering* (2001), pp. 151–162.

[SFWG04] STOKES W. A., FERWERDA J. A., WALTER B., GREENBERG D. P.: Perceptual illumination components: a new approach to efficient, high quality global illumination rendering. *ACM Trans. Graph. 23*, 3 (2004), 742–749.

[Wal91] WALLACE G. K.: The JPEG still picture compression standard. *Commun. ACM 34*, 4 (1991), 30–44.

[Wat87] WATSON A. B.: Efficiency of an image code based on human vision. *Journal of the Optical Society of America A 4*, 12 (Dec 1987), 2401–2417.

[WLC*03] WILLIAMS N., LUEBKE D., COHEN J. D., KELLEY M., SCHUBERT B.: Perceptually guided simplification of lit, textured meshes. In *ACM I3D'03* (2003), pp. 113–121.

[WPG02] WALTER B., PATTANAIK S. N., GREENBERG D. P.: Using perceptual texture masking for efficient image synthesis. *Comput. Graph. Forum 21*, 3 (2002).

[YC04] YUAN X., CHEN B.: Illustrating surfaces in volume. In *Proceedings of Joint IEEE/EG Symposium on Visualization* (2004), pp. 9–16.

[ZDL02] ZENG W., DALY S., LEI S.: An overview of the visual optimization tools in JPEG 2000. *Signal Processing: Image Communication 17* (Jan 2002), 85–104.

[ZPvBG01] ZWICKER M., PFISTER H., VAN BAAR J., GROSS M.: Surface splatting. In *SIGGRAPH '01* (2001), pp. 371–378.