Perceptually Driven Simplification for Interactive Rendering

David Luebke, Benjamin Hallen
University of Virginia

Abstract

We present a framework for accelerating interactive rendering, grounded in psychophysical models of visual perception. This framework is applicable to multiresolution rendering techniques that use a hierarchy of local simplification operations. Our method drives those local operations directly by perceptual metrics; the effect of each simplification on the final image is considered in terms of the contrast the operation will induce in the image and the spatial frequency of the resulting change. A simple and conservative perceptual model determines under what conditions the simplification operation will be perceptible, enabling imperceptible simplification in which operations are performed only when judged imperceptible. Alternatively, simplifications may be ordered according to their perceptibility, providing a principled approach to best-effort rendering. We demonstrate this framework applied to view-dependent polygonal simplification. Our approach addresses many interesting topics in the acceleration of interactive rendering, including imperceptible simplification, silhouette preservation, and gaze-directed rendering.

1 Introduction

Interactive rendering of large-scale geometric datasets continues to present a challenge for the field of computer graphics. Despite tremendous strides in computer graphics hardware, the growth of large-scale models continues to outstrip our capability to render them interactively. A great deal of research has therefore focused on algorithmic techniques for managing the rendering complexity of these models. *Polygonal simplification* algorithms offer a powerful tool for this task. These methods, also known as *level of detail* or *LOD* techniques, hinge on the observation that most of the complexity in a detailed 3-D model is unnecessary when rendering that model from a given viewpoint. These methods simplify small, distant, or otherwise unimportant portions of the scene, reducing the rendering cost while attempting to retain visual fidelity. Visual fidelity has traditionally been measured using geometric criteria. Often, however, the most important measure of fidelity is not geometric but perceptual: does the simplification *look* like the original?

We describe an LOD framework guided directly by perceptual metrics. These metrics derive from the *contrast sensitivity function* or *CSF*, a simple measure of low-level perceptibility of visual stimuli. Testing local simplification operations against a model of the CSF provides a principled approach to the fidelity/performance tradeoff. Our approach addresses several interesting problems in regulating level of detail:

- **Imperceptible simplification:** We evaluate simplification operations by the “worst-case” contrast and spatial frequency they induce in the image, and apply only those operations judged imperceptible. We show that the resulting simplified model is indistinguishable from the original.

- **Best-effort simplification:** Often we wish to render the best image possible within time or polygon constraints. Ordering simplification operations according to the viewing distance at which their effect on the image becomes perceptible provides a principled framework for simplifying to a budget.
- **Silhouette preservation:** Silhouettes have long been recognized as visually important, but how important? Our model quantifies silhouette importance by accounting for their increased contrast, and preserves them accordingly.
- **Gaze-directed rendering:** If the system can monitor the user's gaze, the image may be simplified more aggressively in the visual periphery. We can extend our model to incorporate *eccentricity*, or the falloff of visual acuity in the periphery.

Our framework applies to any rendering system based on hierarchical approximations, such as polygonal mesh reduction, texture-based imposters, and some forms of image- and point-based rendering. In this paper, we explore the application of this framework to view-dependent polygonal simplification. Our key contribution is a technique for evaluating the worst-case perceptibility of local simplification operations, each removing a few polygons from the mesh, according to the contrast and spatial frequency they induce.

2 Background and Previous Work

2.1 The Contrast Sensitivity Function

A large body of perceptual psychology literature focuses on the perceptibility of visual stimuli. The simplest relation established in this literature is *Weber's law*, which predicts the minimum detectable difference in luminance between a test spot on a uniform visual field. At daylight levels, this threshold difference in luminance increases linearly with background luminance. Interesting scenes are not uniform, however, but contain complex frequency content. Outside a small frequency range, the threshold sensitivity predicted by Weber's law drops off significantly. Many perception studies have therefore examined the perceptibility of *contrast gratings*,

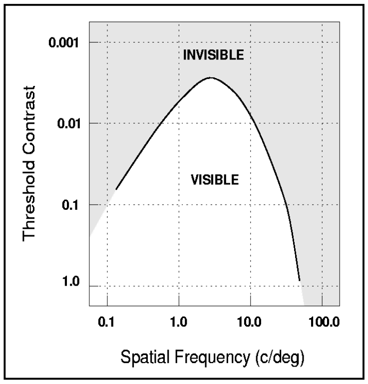

Figure 1: The *contrast sensitivity function* measures the perceptibility of visual stimuli (sinusoidal gratings) in terms of their contrast and spatial frequency (cycles per degree). Courtesy Martin Reddy.

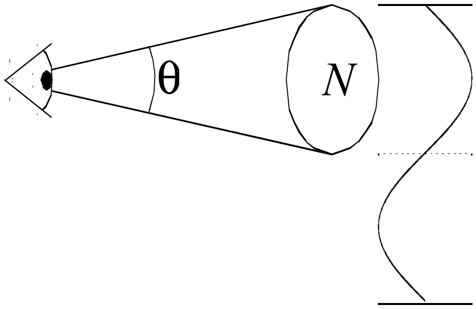

Figure 2: The lowest spatial frequency that can be affected by a node spanning θº of visual arc has one cycle per 2θº.

sinusoidal patterns that alternate between two extreme luminance values L_{max} and L_{min} [3]. Contrast grating studies use *Michaelson contrast*, defined as $(L_{max} – L_{min}) / (L_{max} + L_{min})$, and *spatial frequency*, defined as the number of cycles per degree of visual arc. The *threshold contrast* at a given spatial frequency is the minimum contrast that can be perceived in a grating of that frequency, and *contrast sensitivity* is defined as the reciprocal of threshold contrast. The *contrast sensitivity function* (CSF) plots contrast sensitivity against spatial frequency, and so describes the range of perceptible contrast gratings [Figure 1].

Of course, interesting images are more complex than the simple sinusoidal patterns used in contrast gratings. To a first approximation, however, perceptibility of complex signals can be determined by decomposing a signal into sinusoidal components using Fourier analysis [4]. In particular, if no frequency component of a signal is perceptible, the signal will not be perceptible.

The CSF predicts the maximum perceptibility of a stationary grating at the center of view. Other factors can lower contrast sensitivity further, including *eccentricity*, or angular distance from the direction of gaze. The *fovea* is the region of highest sensitivity on the retina, occupying the central 1^{o} or so of vision. *Visual acuity*, measured as the highest perceptible spatial frequency, is significantly lower in the visual periphery than at the fovea [25]. Extending our perceptual model to incorporate eccentricity lets us predict peripheral visibility for gaze-directed rendering.

2.2 Perceptually Based Offline Rendering

Many researchers have worked on perceptually based rendering algorithms, such as Walter et al [29], Bolin and Meyer [2], and Ramasubramanian et al [23]. The latter two both include good surveys of the field. These algorithms take advantage of the limitations of human vision to avoid rendering computation where the result will be imperceptible. Unlike our work, which seeks to accelerate interactive rendering, almost all previous perceptually based rendering approaches have addressed realistic offline rendering approaches such as ray and path tracing. Since image creation times in such approaches are typically measured in seconds or minutes, these algorithms use sophisticated perceptual models that are costly to evaluate by interactive rendering terms. For example, Ramasubramanian et al take several seconds to evaluate a 512x512 image. This is clearly unsuitable for interactive rendering, which measures frame time in milliseconds.

2.3 Perceptually Based LOD Selection

Regulating scene complexity by simplifying small or distant objects was first proposed in Clark's seminal 1976 paper [5], and several recent surveys examine the current state of the art [10][17][22]. The basic approach described by Clark remains the most common approach today: create several versions of each object, at progressively coarser levels of detail (called *LODs*), and choose at run-time which LOD will represent the object.

Comparatively few systems have attempted to guide this process with explicit perceptual metrics. Funkhouser and Sequin used a cost-benefit estimate to pick the best levels of detail within a specified time budget [9]. Their system used ad hoc weighting factors to account for eccentricity and *velocity*, the speed at which the image of an object moves across the retina. Ohshima et al described a system for gaze-directed stereoscopic rendering [21]again using heuristic models of eccentricity, velocity, and convergence to guide selection of precomputed LODs.

Reddy was the first to attempt an LOD selection system guided throughout by a principled perceptual model [24]. Using images rendered from multiple viewpoints, Reddy analyzed the frequency content of objects and their LODs, A model of the visual acuity, defined as highest perceptible spatial frequency, guided LOD selection: if a high-resolution and a low-resolution LOD differed only at frequencies beyond the visual acuity of the viewer, the system used the low-resolution LOD. Scoggins et al analyzed the frequency content more thoroughly, transforming a prerendered reference image to frequency space [28]. Scoggins et al then applied the CSF as a modulation transfer function and used the resulting mean-squared error to decide which LOD was appropriate.

2.4 View-Dependent Polygonal Simplification

One difficulty with all these approaches is their reliance on a few discrete levels of detail to represent each object. This limits the degree to which perceptual metrics can be applied, since the entire object must be simplified uniformly. For example, silhouette details tend to be more perceptible than interior details because of higher contrast, so the entire object must be treated as if it were on the silhouette. *View-dependent* simplification methods offer a solution. Rather than calculating a series of static LODs, view-dependent systems build a data structure from which a desired level of detail may be extracted at run time. Objects in a view-dependent algorithm may span multiple resolutions; for example, portions of the object under the viewer's gaze can be represented at higher fidelity than portions in the peripheral vision.

Several researchers have proposed view-dependent algorithms, including Hoppe, Luebke, and Xia [11][16][30]. These algorithms use a hierarchy of *vertex merge* operations that can be applied or reversed at run-time. Our chief contribution is a method for evaluating the perceptibility of a vertex merge. We have implemented our system using *VDSlib*, a public-domain library that allows users to plug in custom callbacks for building, culling, simplifying, and rendering the model [19]. We first augment VDSlib with some perceptual data (described below), then at run time, our callback examines possible simplifications, using contrast, spatial frequency, and possibly eccentricity to decide which vertices VDSlib should merge. Before describing the details of this process, we briefly review the VDSlib algorithm and notation.

The main data structure of VDSlib is the *vertex tree*, a hierarchical clustering of vertices. Leaf nodes of the tree represent a single vertex from the original model; interior nodes represent multiple vertices clustered together, and the root node represents all vertices from the entire model, merged into a single cluster. In VDSlib parlance, a node **N** *supports* a vertex **V** if the leaf node associated with **V** descends from **N**. Similarly, **N** *supports* a triangle **T** if it supports one or more of the corner vertices of **T**. The set of triangles in the model supported by a node is called the *region of support* of the node.

Each node stores a representative vertex called the *proxy*. For leaf nodes, the proxy is exactly the vertex of the original model that the node represents; for interior nodes, the proxy is typically some average of the represented vertices. *Folding* a node merges all of the vertices supported by that node into the node's single proxy vertex. In the process, triangles whose corners have been merged together are removed, decreasing the overall polygon count of the scene. Since folding a node is the core simplification operation of VDSlib, to apply our perceptual framework we must evaluate the contrast and spatial extent of the change in the rendered image induced by a fold.

3 Overview of our approach

Our main contribution is a way to map the change resulting from a local simplification operation to a *worst-case* contrast grating, meaning a grating with the most perceptible possible combination of contrast and frequency induced by the operation. This gives us a bound on the perceptibility of that simplification operation. For example, for imperceptible simplification we apply only those operations whose corresponding gratings we would not expect to be visible, while for best-effort simplification we order the simplification operations according to the perceptibility of their gratings.

3.1 Determining the Worst Case

We wish to characterize the frequency and contrast induced in the rendered image by a simplification operation, but this induced change will generally have a non-trivial spectrum, with multiple frequencies present at different amplitudes. Since the CSF is non-linear, it does not obviously follow that the frequency component with the greatest amplitude is the most perceptible. Performing a Fourier transform of the image in the neighborhood of each local operation and modulating the resulting frequencies by the CSF could evaluate the most perceptible frequency, but this is clearly too expensive. We argue below that if all induced frequencies were present at equal amplitudes, the *lowest* frequency would be the most perceptible. Furthermore, a conservative estimate

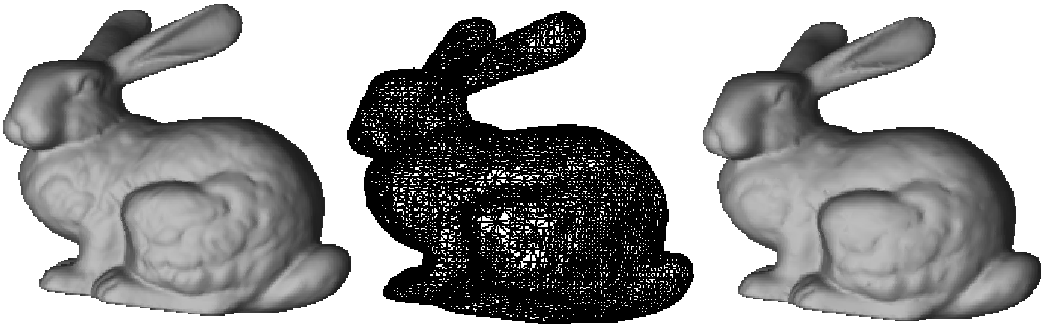

Figure 3: The original Stanford Bunny model (69,451 faces) and a simplification by our perceptually driven system (29,866 faces). In this view the user's gaze is 29° from the center of the bunny…equivalent to looking at the top of this page from a distance of 29 cm. Note that the silhouette is well preserved, along with strong interior details (the line of the haunch, the shape of the eye, etc.) while subtle bumps on the surface are simplified. The bunny and other models used here are courtesy of the Stanford 3-D Scanning Repository.

of the induced contrast establishes a lower bound on the amplitude of any frequency component. Thus our key observation: the perceptibility of a change induced by simplification can be conservatively equated to the perceptibility of the *lowest frequency* induced by that change, at the *maximum contrast* induced by that change.

To show this, we make some conservative simplifying assumptions. We observe that the peak contrast sensitivity occurs at approximately 2-4 cycles per degree, and that most local simplification operations on a complex model only affect much higher frequencies. We therefore assume that contrast at lower spatial frequencies is more perceptible than at higher frequencies, and ensure this assumption holds by clamping our worst-case frequency to be no lower than the point of peak sensitivity. The minimum frequency component of a region in the image spanning n degrees of the user's angular field of view is one cycle per $2n$ degrees. Put another way, the maximum wavelength needed to represent a region of the image is twice the maximum spatial extent of that region [Figure 2]. Consequently, we can reduce finding the worst-case frequency induced by a simplification to finding the screen-space extent of the affected region.

For the worst-case contrast, we determine a bound on the maximum change in luminance among all the pixels affected by the simplification. The worst-case contrast of a simplification operation is thus the maximum contrast between an image of the affected region at full resolution and an image of the region simplified. For 3-D models, there are two basic cases:

- The entire affected region lies interior to a surface that entirely faces the viewer. This is the simplest case: the contrast between the original region and the folded region is completely determined by the luminance of the local surface before and after the fold.
- The affected region includes a silhouette or visual contour. This expands the possible contrast incurred by the simplification to include the portion of the scene *behind* the affected region, since simplifying the surface may expose a very bright or very dark feature occluded before simplification.

Consequently, silhouette regions of the object are simplified less aggressively—exactly the behavior we should expect in a perceptually driven simplification algorithm. Note, however, that even at these higher contrast levels silhouette regions can still be simplified if they represent very fine details (high spatial frequencies) or are in the viewer's peripheral vision (high eccentricity).

3.2 An Empirical Perceptual Model

Many researchers have characterized the contrast sensitivity function. In early work, Kelly derived an abstract relationship for the perceptibility of sinusoidal gratings over a narrow range [13]. A broader range was described by the equation of Mannos and Sakrison [20]. More recent and accurate CSF models, such as the models given by Barten [1] and Daly [7], are used in current advanced global illumination algorithms [23][2]. Modern perceptual theory attributes the shape of the CSF to the combined response curves of multiple bandpass mechanisms in the visual system, each processing only a small range of the visible spatial frequency spectrum. This multiscale visual processing can be emulated with a Laplacian pyramid for spatial decomposition [15]. Current perceptual rendering techniques also account for *contrast*

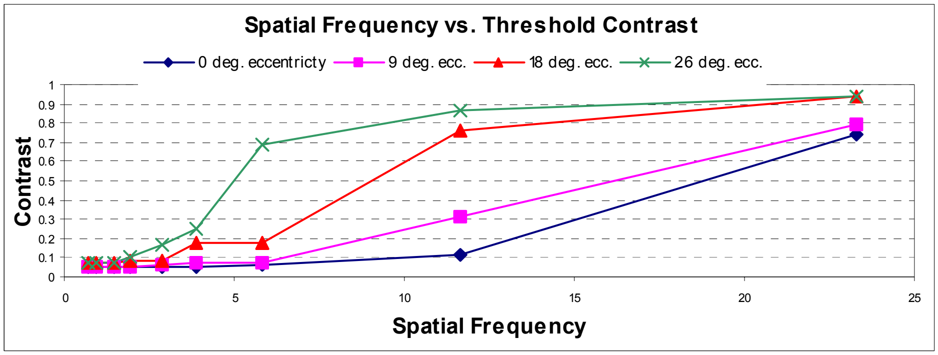

Figure 4: We use a simple model of contrast sensitivity based on an empirical calibration procedure. Shown here are results from one user's calibration.

masking, which represents the visual system's decreased contrast sensitivity in the presence of strong patterns. This can further increase the allowable error in an image [23][2][8].

Unfortunately, these sophisticated perceptual models, which employ the latest advances in understanding perception, are far too costly for the interactive framework we propose. In our framework, thousands of simplification operations must be considered every second, leaving less than a millisecond to evaluate the induced contrast and frequency. Clearly, we must forego the state-of-the-art perceptual models used in current global illumination work for a simple, fast, and conservative model.

We have had promising results using a simple mathematical model of the CSF, namely Rushmeier's simplification of Daly's equation [26]. However, for the results shown here we chose to take an empirical approach that allows us to achieve simplicity and speed, while still accounting for real-world factors (such as ambient light) that affect perception. Recall our hypothesis: a simplification operation, mapped to a "worst-case" contrast grating, can be performed imperceptibly if that grating would not be perceptible. We build our perceptual model directly from contrast grating tests performed under the same conditions—room illumination, monitor, etc—under which our final system will run. A calibration procedure tests the ability of a user to detect contrast gratings, recording threshold contrast over a wide range of spatial frequency and eccentricity. We then build a lookup table from the resulting CSF curves and use linear interpolation at runtime to determine whether the user can perceive a given contrast at a given spatial frequency and eccentricity. As mentioned in Section 3.1, we ensure conservative behavior by clamping threshold contrast below the frequency of peak sensitivity.

This model could certainly be improved, but we chose to focus on developing a framework for driving interactive rendering with a perceptual model, rather than on developing the model itself. Our empirical model is simple to implement and works well in practice. Figure 4 shows example CSF curves determined from a typical calibration procedure. Note that we must calibrate the monitor to map OpenGL intensities to luminance. We used a photometer, but simple gamma correction would suffice if less precision were required.

3.3 Evaluating the fold operation

Folding a node in VDSlib can affect the rendered image in complex ways. As the vertices and triangles supported by the node merge and shift, features in the image may shrink, stretch, or disappear completely. Shifting triangles on the silhouette may expose previously occluded features. To analyze the effect of folding a node, we should consider all of these changes. One possibility, recently demonstrated by Lindstrom and Turk for static LOD generation, is to render the scene before and after the operation and analyze the resulting images [14]. At present, however, the requisite rendering and image processing appears too expensive for dynamic simplification. Instead, we want a conservative worst-case bound on the changes in the image caused by folding the node. As discussed, this worst case bound can be reduced to finding the minimum frequency and the maximum contrast.

Spatial Frequency: Estimating Extent

Recall that the minimum frequency induced by a simplification is determined by the spatial extent of the resulting change in the image. Notice that features in the image affected by a fold consist of triangles connecting vertices involved in the fold. The largest feature that can be removed or exposed by geometric distortion upon folding a node is therefore constrained by the distance vertices move during the fold. Thus, the problem of computing the minimum frequency induced by folding a node reduces to computing the screen-space extent of all vertices supported by the node.[1] We use bounding spheres to estimate this extent, associating with each node a tight-fitting sphere that contains all vertices in the node's region of support. The angular extent of these bounding spheres, as seen from a given viewpoint, can be calculated very quickly. The minimum frequency affected by folding a node is then one cycle per two degrees of angular extent spanned by the node's bounding sphere [Figure 2].

Contrast: Estimating Intensity Change

Determining the precise contrast induced by folding a node would be as expensive as rendering the unfolded geometry. Instead, we want a conservative lower bound, which could be computed several ways. For simplicity, we currently assume pre-lit Gouraud-shaded meshes, and obtain a conservative bound by comparing the intensities of all the vertices the node supports in the original model with the intensities of the vertices in the simplified surface. The greatest difference between the intensities of the surface vertices before folding and after folding bounds the maximum contrast between the simplified surface and the original surface, since in a Gouraud-shaded model extremes of intensity always occur at the vertices. This test may overestimate the contrast induced by folding, but will not underestimate it.

When the node's region of support includes a silhouette, we must be even more conservative. Lacking knowledge about what lies behind the region, we must assume the worst: moving a silhouette edge might expose the darkest or brightest object in the scene, including the background. Hence we must compare the range of vertex

---

[1] Technically, this holds when the model is flat shaded; for Gouraud-shaded models, adjacent vertices should also be included. However, we have not found this necessary in practice.

intensities of the node's region of support against the brightest and darkest intensities in the scene, and use the maximum possible difference in intensity for calculating the contrast induced by the fold.

Determining Silhouette Nodes

Since nodes affecting silhouette edges must be treated differently, we require an efficient method for identifying such nodes. For a given view, we define *silhouette node*s as those nodes supporting both front-facing and back-facing triangles in the original mesh. One possibility would use the normal cone hierarchies of Johnson and Cohen [12], but we currently use a bitwise approach inspired by the rapid backface culling technique of Zhang and Hoff [31]. We quantize the Gauss sphere of normal space to a *normal cube* whose faces are tiled into cells, and store a per-node *normal mask*, or bit vector representing the normals of all its supported triangles. The silhouette test may then be implemented with simple bitwise operations. This technique is fast and accurate, though it requires more storage (we use 48 bytes/node) than the normal cone hierarchies.

Putting It All Together: Imperceptible Simplification

Given these elements, imperceptible simplification is easily implemented. A VDSlib traversal visits each node in the hierarchy top-down, applying our custom simplification criterion as a callback at each node. The callback evaluates the worst-case frequency based on the screen-space size of the node, and then looks up the threshold contrast for that frequency in our empirical CSF model. If the contrast induced by folding is less than the threshold contrast, the callback allows VDSlib to fold the node, otherwise the node is left unfolded and traversal continues.

3.4 Perceptually Guided Best-Effort Rendering

We can also use our model for best-effort simplification. VDSlib supports *triangle budget simplification*, which lets the user specify how many triangles the scene should contain. Using a user-specified error metric, VDSlib applies a greedy algorithm to

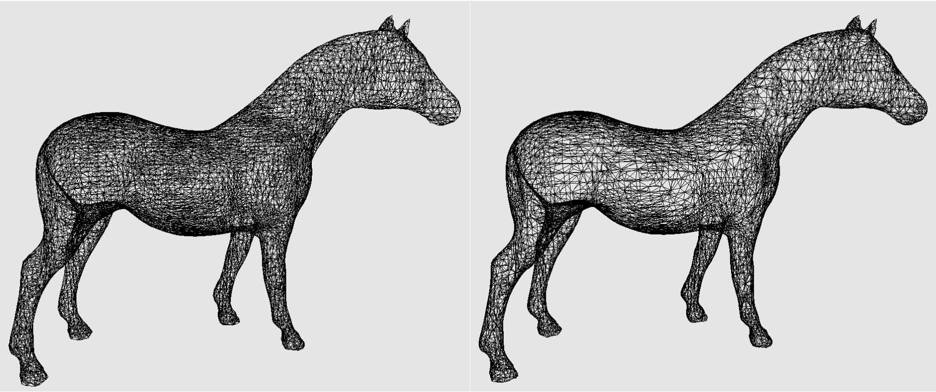

Figure 5: Perceptually driven best-effort simplification. Both images show the horse model (originally 96,966 faces) reduced to 18,000 faces using triangle budget rendering in VDSlib. Left, the default VDSlib error metric uses screenspace node size, leading to unnecessarily even tesselation. Right, our perceptually driven metric uses fewer polygons in interior and low-contrast regions.

minimize the total error induced by all folded nodes, while staying within this triangle budget constraint. We must therefore generate a sound perceptual measure of the error introduced by folding a node. The key is to recast our metric for evaluating the perceptibility of fold operations: rather than a binary perceptible/not perceptible decision, we need a scalar to express *how* perceptible a fold operation could be.

Unfortunately, the CSF provides only threshold information, and cannot be used directly to evaluate suprathreshold perceptibility. In other words, the CSF can tell us whether a fold operation is perceptible, but cannot tell us which of two perceptible folds is more objectionable. We therefore chose to cast the question in terms of distance: how far would the viewer have to be from the screen before the node could be folded imperceptibly? The answer can be computed from our current perceptual model, in effect by inverting our lookup tables. Rather than computing the spatial frequency of a node and looking up the threshold contrast at which folding is perceptible, we use the precomputed contrast induced by folding and look up the threshold spatial frequency. From this we can compute the distance at which folding the node would be perceptible, and sort nodes to be folded according to this distance. We then order folds based on the viewing distance at which they become perceptible and stop when the budget is reached. This provides an intuitive physical measure of the fidelity achieved, since the system can report the distance at which a simplification should be imperceptible.

3.5 Results

All results given are on an 866 MHz Pentium III computer with NVidia GeForce^2 graphics. Figure 3 shows a model simplified imperceptibly while accounting for eccentricity. Since we are guaranteeing imperceptible simplification, the reductions in polygon count may seem modest. However, these results and our user study (below) clearly show that perceptually driven simplification can reduce model complexity without visual effect.

Perceptually driven best-effort rendering may be of more use to many 3-D applications. Figure 5 compares our results to VDSlib's built-in triangle budget rendering, which orders fold operations only by screen-space size of the node. Note that the perceptually driven algorithm preserves more triangles near silhouettes, and simplifies more aggressively in regions of low contrast.

We have performed a preliminary user study to determine whether our simplifications are indeed imperceptible from the original model. The study tested whether 4 subjects in 200 trials could perceive any difference between a rendering of a full-resolution model and a rendering of a model simplified with our algorithm. The study, which we do not describe in detail here, confirmed that subjects' ability to discern the simplification was no better than chance. For a full description of the study, please see our technical report [18].

4 Summary And Discussion

Perceptually guided interactive rendering is a broad and difficult topic. Our system shows the feasibility and potential of imperceptible view-dependent simplification, but many avenues for further research remain. Below we summarize our contribution and results, and address what we see as pressing and interesting directions for future work.

4.1 Summary

We have demonstrated a novel approach to reducing model complexity that is directly driven by perceptual criteria. Our principle contribution is a practical framework for perceptually guided interactive rendering that equates local simplification operations to worst-case contrast gratings whose perceptibility we can evaluate. We have demonstrated this framework in the context of view-dependent polygonal simplification. Our approach addresses several interesting problems, including silhouette preservation and imperceptible simplification. An optional gaze-directed component uses eye tracking to obtain further simplification by reducing fidelity in the viewer's peripheral vision.

4.2 Ongoing and Future Work

We have demonstrated our perceptual framework applied to view-dependent polygonal simplification, but the framework also applies to many other rendering schemes. We are experimenting with a perceptually driven version of the *Qsplat* point-rendering system by Rusinkiewicz and Levoy [27], which provides a completely different model representation and rendering paradigm. We believe this is an excellent testimonial to the flexibility and generality of our framework.

The current system is far from perfect; we see this work as the first step rather than the last word in perceptually guided interactive rendering. Our chief problem: the system is overly conservative. In practice, we find that our models could be reduced two to three times further in polygon count without perceptible effect. We attribute this primarily to our highly conservative estimate of spatial frequency, and are exploring more accurate ways to compute the induced frequencies.

Incorporating dynamic lighting into the contrast calculation is an obvious extension; this should be quite possible given the node normal masks. Incorporating texture mapping is an exciting area for future work, and could both increase and decrease the amount of simplification possible. The distortion of a texture on a simplified surface being simplified could increase the perceptibility of the simplification, but the frequency content of the texture could potentially be analyzed in a preprocess to account for visual masking effects that would decrease perceptibility of simplification. We are investigating integrating the texture deviation metric of Cohen's *appearance-preserving simplification* [6] to account for these factors. Like other perceptually driven rendering algorithms (e.g., [23][2]) we model perceptibility of static stimuli; we believe that the field as a whole needs to begin incorporating measures of *temporal contrast sensitivity* to address possible flicker artifacts in interactive or animated rendering. Finally, we would like to exploit the reduced visual acuity caused by velocity across the retinal field, which Reddy's work suggests provides a promising opportunity for further simplification [24].

5 References

[1] Barten, Peter. "The Square-Root Integral", In *Human Vision, Visual Processing, and Digital Display*, vol. 1077, Proceedings SPIE (1989)

[2] Bolin, Mark. and G. Meyer. "A Perceptually Based Adaptive Sampling Algorithm", *Computer Graphics*, Vol. 32 (SIGGRAPH 98).

[3] Campbell, F., Gubisch, R. "Optical Quality of the Human Eye", *Journal of Physiology* 186.

[4] Campbell, F.W. and Robson, J.G. “An Application of Fourier Analysis to the Visibility of Contrast Gratings”, *Journal of Physiology*, 187 (1968)

[5] Clark, James H. “Hierarchical Geometric Models for Visible Surface Algorithms,” *Communications of the ACM*, Vol. 19, No 10, pp 547-554.

[6] Cohen, J, M. Olano, and D. Manocha. “Appearance-Preserving Simplification,” *Computer Graphics*, Vol. 32 (SIGGRAPH 98).

[7] Daly, S. “Visible differences predictor: An algorithm for the assessment of image fidelity,” *Digital Images and Human Vision* (A. Watson, ed.), pp 179--206, MIT Press (1993).

[8] Ferdwada, James, S. Pattanaik, P. Shirley, and D. Greenberg. “A Model of Visual Masking for Realistic Image Synthesis”, *Computer Graphics*, Vol. 30 (SIGGRAPH 96).

[9] Funkhouser, Tom, and C. Sequin. “Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments”, *Computer Graphics*, Vol. 27.

[10] Heckbert, Paul, and M. Garland. “Survey of Polygonal Surface Simplification Algorithms”, SIGGRAPH 97 course notes (1997).

[11] Hoppe, Hughes. “View-Dependent Refinement of Progressive Meshes”, *Computer Graphics*, Vol. 31 (SIGGRAPH 97).

[12] Johnson, David, and E. Cohen. “Spatialized Normal Cone Hierarchies”, *Proceedings ACM Symposium on Interactive 3D Graphics* (2001).

[13] Kelly, D.H. “Spatial Frequency Selectivity in the Retina”, *Vision Research*, 15 (1975).

[14] Lindstrom, Peter. and Turk, G. “Image-Based Simplification”, *ACM Transactions on Graphics*, July 2000 (2000).

[15] Lubin, Jeffery. “A Visual Discrimination Model for Imaging System Design and Evaluation”, *Vision Models for Target Detection and Recognition*, World Scientific (1995).

[16] Luebke, David, and C. Erikson. “View-Dependent Simplification of Arbitrary Polygonal Environments”, *Computer Graphics*, Vol. 31 (SIGGRAPH 97).

[17] Luebke, David. “A Developer’s Survey of Polygonal Simplification Algorithms”, *IEEE Computer Graphics & Applications* (May 2001). See tech report CS-99-07, U of Virginia.

[18] Luebke, David, and B. Hallen. “Perceptually-Driven Interactive Rendering”. Technical report CS-2001-01, University of Virginia (2000).

[19] Luebke, David. See http://vdslib.virginia.edu.

[20] J. L. Mannos, D. J. Sakrison, ``The Effects of a Visual Fidelity Criterion on the Encoding of Images'', *IEEE Transactions on Information Theory*, pp. 525-535, Vol. 20, No 4, (1974).

[21] Oshima, Toshikazu, H. Yamammoto, and H. Tamura. “Gaze-Directed Adaptive Rendering for Interacting with Virtual Space”, *Proceedings of VRAIS 96* (1996).

[22] Puppo, Enrico, and R. Scopigno. “Simplification, LOD and Multiresolution—Principles and Applications”, *Eurographics '97 Tutorial Notes*, PS97 TN4 (1997).

[23] Ramasubramanian, Mahesh, S. Pattanaik, and D. Greenberg. “A Perceptually Based Physical Error Metric for Realistic Image Synthesis”, *Computer Graphics*, Vol. 33.

[24] Reddy, Martin. “Perceptually-Modulated Level of Detail for Virtual Environments”, Ph.D. thesis, University of Edinburgh, 1997.

[25] Rovamo, J. and Virsu, V. “An Estimation and Application of the Human Cortical Magnification Factor”, *Experimental Brain Research*, 37 (1979)

[26] Rushmeier, H., G. Ward, C. Piatko, P. Sanders, and B. Rust. “Comparing Real and Synthetic Images: Some Ideas About Metrics,” In *Rendering Techniques '95*, pp 82-91, Springer-Verlag (1995).

[27] Rusinkiewicz, S. and Levoy, M. “QSplat: A Multiresolution Point Rendering System for Large Meshes”, *Computer Graphics*, Vol. 34 (SIGGRAPH 2000).

[28] Scoggins, Randy, R. Machiraju, and R. Moorhead. “Enabling Level-of-Detail Matching for Exterior Scene Synthesis”, *Proceedings of IEEE Visualization 2000* (2000).

[29] Walter, Bruce, P. M. Hubbard, P. Shirley, and D. Greenberg. “Global Illumination using Local Linear Density Estimation”, *ACM Transaction on Graphics* (1997).

[30] Xia, Julie and Amitabh Varshney. “Dynamic View-Dependent Simplification for Polygonal Models”, *Visualization 96*.

[31] Zhang, Hansong, and K. Hoff. “Fast Backface Culling Using Normal Masks”, *Proceedings of ACM Symposium on Interactive 3D Graphics* (1997).