Perceptually Driven Simplification Using Gaze-Directed Rendering

David Luebke, Benjamin Hallen, Dale Newfield, and Benjamin Watson

University of Virginia Technical Report CS-2000-04

1 ABSTRACT

We present a unique polygonal simplification method grounded in rigorous perceptual science. Local simplification operations are driven directly by perceptual metrics, rather than the geometric metrics common to other algorithms. The effect of each operation 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. Equations derived from psychophysical studies determine whether the simplification operation will be perceptible; the operation is performed only if its effect is judged imperceptible. To increase the range of simplification permitted, we incorporate *gaze-directed rendering*. A commercial eye tracker monitors the direction of the user's gaze, allowing the image to be simplified more aggressively in the periphery than at the center of vision. Our perceptual model addresses many interesting topics in polygonal simplification, including gaze-directed rendering, silhouette preservation, and imperceptible simplification. We describe two user studies to evaluate our model, and address the shortcomings as well as the potential of our approach.

2 INTRODUCTION

Interactive rendering of large-scale geometric datasets continues to present a challenge for the field of computer graphics. Such interactive visualization is an enabling technology for many far-flung fields, ranging from scientific and medical visualization to entertainment, architecture, military training, and industrial design. 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 focused on algorithmic techniques for managing the geometric complexity of these models. *Geometric simplification* methods offer a powerful tool for this task. Geometric simplification, often referred to as *level of detail* or *LOD*, hinges on the observation that most of the complexity in a very 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, however, has traditionally been difficult to quantify, so most simplification algorithms settle for geometric measures of quality. For example, fidelity of the simplified surface may be assumed to vary with the distance of that surface from the original mesh, or with the volume of distortion created by the simplification. Such metrics are useful for certain CAD applications, such as finite element analysis, and for certain medical and scientific visualization tasks, such as co-registering surfaces or measuring volumes. Probably the most common purpose of simplification, however, is to speed up rendering of complex databases. For this purpose, the most important measure of fidelity is not geometric but perceptual: does the simplification *look* like the original?

In this paper, we present a geometric simplification algorithm guided by perceptual metrics. These metrics derive from a large body of cognitive psychology literature on the perceptibility of visual stimuli, classifying those stimuli according. The motivating question driving our work has been:

> *Can we generate simplifications that we can rigorously assert are indistinguishable from the original model to the typical observer?*

Since our goal of perceptual rigor necessarily limits the degree of simplification, the resulting algorithm incorporates an unusual technique: *gaze-directed simplification*. We track the eye gaze of the viewer, using this knowledge to degrade the scene more aggressively in the viewer's peripheral vision than at the center of their gaze.

2.1 Contribution

Gaze-directed simplification is not new, but we believe our approach is the first to unify rigorous perceptually based metrics with a view-dependent framework powerful enough to achieve a useful reduction in model complexity. Unlike previous approaches, we apply perceptual metrics directly to the 3-D model, evaluating the effect of each local geometric simplification operation. Finally, we believe that our results are the first to use actual eye tracking rather than head tracking to monitor the user's gaze.

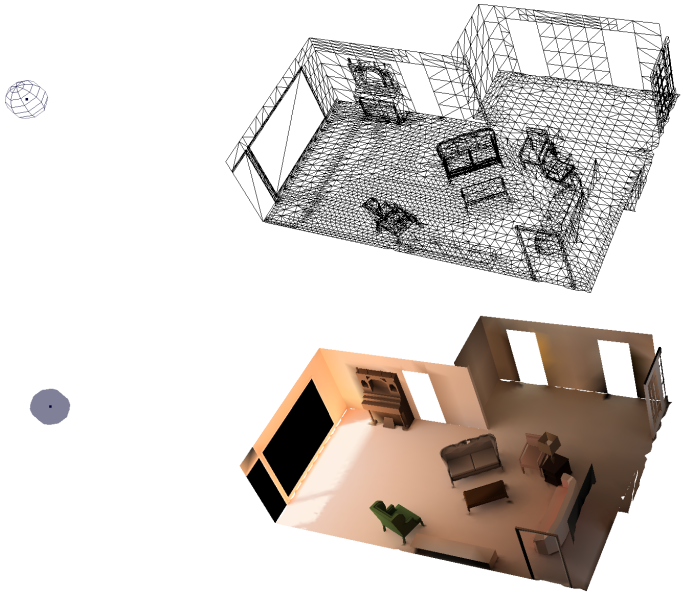

Figure 1: A radiositized architectural scene containing 38,149 polygons rendered with perceptually driven simplification. With the user's current gaze point (indicated by the blue sphere), the model shown here contains only 11,286 polygons. Our perceptual metrics ensure that when viewed as a full-screen image at 46° field of view, this simplification is imperceptible to the typical observer.

3 PREVIOUS WORK

3.1 Polygonal Simplification

Regulating scene complexity and rendering time with geometric simplification of small or distant objects was first proposed in Clark's seminal 1976 paper, and flight simulators have long made use of this technique [Clark 76, Cosman 81]. Their basic approach is still the most common approach today: create several versions of each object in a preprocess, at progressively coarser levels of detail. At run-time, the system chooses which LOD will represent the object, usually based on distance. Because interactive rendering hardware specializes in rendering polygonal models, most geometric simplification efforts have focused on *polygonal simplification*. The past decade has seen a flurry of research into polygonal simplification algorithms to automatically generate coarse LODs from a full-resolution original model. Modern algorithms excel in speed, robustness, and fidelity; an excellent survey is given in [Garland 97]. These algorithms have become a common and crucial tool for the interactive graphics developer.

Early work on gaze-directed rendering includes Funkhouser and Sequin's system for dynamic LOD selection [Funkhouser 93]. This system uses a cost-benefit estimate to pick the best levels of detail within a specified time budget. LOD benefit is assigned heuristically, based primarily on an object's screen-space size. Their system also takes into account *motion blur*—governed by the speed at which the image of an object moves across the retina—and *focus*—governed by the distance from the object to the center of the user's gaze. Lacking an eye- or head-tracking system, the user's gaze is assumed to lie in the center of the screen; lacking accurate perceptual models, the effects of motion blur and focus are controlled with sliders set by the user. Though their incorporation was ad-hoc, this important work introduced the notion of perceptually guided metrics based on gaze direction. Following the perceptual literature, we discuss Funkhouser's motion blur and focus below in terms of *velocity* and *eccentricity*, respectively.

Ohshima *et al* described a system for gaze-directed stereoscopic rendering [Ohshima 96]. Although the paper mentions an eye-tracking system in progress, the results were gathered using head-tracked viewing direction to approximate gaze direction. Their system uses eccentricity, velocity, and *fusion*—the convergence of the eyes on a given point, with loss of fidelity for points closer or further away—to guide selection of precomputed LODs. Each of these factors is modeled with an equation, but no description is given for why those particular equations were chosen or how the constants involved were set. Indeed the equations used to model the three perceptual effects, and the method for combining all three effects to choose an LOD, appear to have been determined empirically. Thus their algorithm, while clearly demonstrating the potential of a gaze-directed approach, still employs a fundamentally heuristic model of the visual system.

[Reddy 97] was the first to attempt an LOD selection system guided throughout by a rigorous model of the human perceptual system. Using the psychophysical findings described below, Reddy analyzes the frequency content of objects and their LODs from multiple viewpoints. A model of the *visual acuity*, or highest perceptible spatial frequency, guides LOD selection. If the differences between a high-resolution and a low-resolution LOD occur at frequencies above the visual acuity of the viewer, they are imperceptible and the low-resolution LOD may be used. Working from previous results and from experimentally gathered evidence, Reddy models the decrease in visual acuity with eccentricity and velocity, and makes his decisions on which LOD to use based on that information.

One problem plaguing all these approaches is their reliance on traditional polygonal simplification techniques. As explained above, these techniques precompute levels of detail—progressively coarser versions of an object—to replace that object at run-time. Since the LODs are created in a preprocess, the object must be simplified uniformly; no view-dependent information is available to guide the reduction in detail. This requires a very conservative strategy: find the most perceptible part of the object and treat the entire object at that level of perceptibility. For example, if the user's eye rests on any portion of the object, the system must treat the entire object as if it were under direct scrutiny (probably forcing the system to render the full-resolution object rather than an LOD). By operating on a per-object level, traditional LOD methods have forced previous gaze-directed rendering research to make worst-case decisions that often prevent useful simplification rates.

View-dependent polygonal simplification methods offer a solution. These algorithms depart from the traditional approach: rather than calculating a series of static levels of detail in a preprocess, view-dependent systems build a data structure from which the desired level of detail may be extracted *at run time*. Objects in a view-dependent algorithm may span multiple resolutions, addressing the worst-case problem described above. For example, portions of the object under the viewer's gaze can be represented at higher fidelity than portions near the periphery, and regions of the object moving slowly across the visual field can utilize higher resolution than fast-moving regions. Several researchers have independently proposed view-dependent algorithms, including [Hoppe 97, Luebke 97, Xia 96]. These algorithms share a common feature: each is a hierarchy of *vertex merge* operations that can be applied or reversed at run-time. Our algorithm uses VDS, a framework for view-dependent simplification described in [Luebke 97], and is built on top of *VDSlib*, a public-domain library implementing that framework.

The main data structure of VDS is the *vertex tree*, a hierarchical clustering of vertices. Vertices from the original model are grouped with nearby vertices into clusters, then the clusters are clustered together, and so on. 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 VDS 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 forms its *region of support*.

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 vertices have been merged together are removed from the scene, decreasing the overall polygon count. Precomputing and storing these triangles with the node makes the fold operation fast enough to perform dynamically, enabling run-time simplification based on view-

dependent criteria (for further details, see [Luebke 97]). In this paper, we analyze the visual effect of a fold operation, using perceptual criteria to fold nodes only when the resulting image should be indistinguishable from an image of the full-resolution model.

3.2 Analyzing Perceptibility

Perceptual psychology has a large body of literature on the perceptibility of visual stimuli. This literature forms the foundation of our metrics for evaluating simplification. Many perception studies have examined the perceptibility of *contrast gratings*, sinusoidally varying patterns that alternate between two extreme luminance values. [Campell 66] and [Rovamo 79] showed that the perceptibility of a contrast grating depends on its luminance contrast, spatial frequency, and eccentricity. *Spatial frequency* is defined as number of cycles per degree of visual arc; *eccentricity* is the angular distance from the center of gaze. Of course, most interesting images are more complex than simple sinusoidal patterns. [Campbell 68] found that the perceptibility of complex signals can be determined by decomposing a signal into sinusoidal components using Fourier analysis. In particular, if no frequency component of a signal is perceptible, the signal will not be perceptible.

The *fovea* is the region of the retina of highest sensitivity, occupying the central 1º or so of vision. Many studies have evaluated the perceptibility of contrast gratings within the fovea. This perceptibility may be expressed as a threshold contrast $c_{threshold}$ dependent on spatial frequency. If the contrast of the grating lies below the threshold contrast for the spatial frequency of that grating, the grating is imperceptible. Using a model of the sustained ganglion cell within the eye, [Kelly 75] derived an abstract relationship for the perceptibility of sinusoidal gratings:

$$\frac{1}{\boldsymbol{c}_{\boldsymbol{Threshold}}} = \alpha^2 \boldsymbol{e}^{-\alpha} \qquad \text{[Equation 1]}$$

Here $c_{Threshold}$ represents the threshold contrast and α represents spatial frequency. Empirical studies of contrast sensitivity report similar functions over specific ranges of spatial frequencies [Savoy 75]. Note that Kelly's abstract model describes the relationship between contrast threshold and spatial frequency; a scaling factor δ must be incorporated to account for variations in luminance and viewing conditions between perceptual studies:

$$\frac{1}{\boldsymbol{c}_{\boldsymbol{Threshold}}} = \delta\alpha^2 \boldsymbol{e}^{-\alpha} \qquad \text{[Equation 2]}$$

Visual acuity, measured as the highest perceptible spatial frequency, is lower in the visual periphery than at the fovea. This relationship between visual acuity and eccentricity is characterized as the *cortical magnification factor*. [Rovamo 79] studied published data to empirically determine the human cortical magnification factor M, which actually varies according to the region of the retina. At the most sensitive portion of the retina, the *temporal region*, they found that the visual acuity at an eccentricity $\boldsymbol{E}$ (measured here in degrees) is proportional to the visual acuity at the fovea according to M_t:

$$\boldsymbol{M}_{\boldsymbol{t}} = \frac{1}{1+0.29\boldsymbol{E}} \qquad \text{[Equation 3]}$$

The velocity of a feature across the visual field also affects the perceptibility of detail, which leads to motion blur. [Koenderink 78] found that the perceptibility of moving contrast gratings may be related to that of static contrast gratings by a scaling function. Kelly conducted empirical studies; his findings relate velocity to contrast threshold and spatial frequency by the equation:

$$\frac{1}{\boldsymbol{c}_{\boldsymbol{Threshold}}} = \left[6.1 + 7.3\left|\log_{10}(\boldsymbol{v}/3)\right|^3\right]\boldsymbol{v}\alpha^2\boldsymbol{e}^{-\frac{2\alpha(\boldsymbol{v}+2)}{45.9}} \qquad \text{[Equation 4]}$$

Here v is the velocity of a contrast grating measured in degrees per second across the visual field. Experiments by [Reddy 97] confirmed the abstract relationship within Kelly's equation, but found different constants.

To avoid introducing more than one variable in our perceptual metrics, we elected to concentrate on peripheral degradation due to eccentricity, and currently ignore the additional degradation introduced by retinal velocity. However, utilizing and integrating retinal velocity is an obvious next step for future research.

3.3 Overview of the Approach

Our goal is to analyze the polygonal simplification process from a rigorous perceptual framework; the underpinnings of our analysis are the contrast grating studies described above. These studies classify the perceptibility of image features according to their *contrast* and their *spatial frequency*. Our approach is to map the change resulting from a local simplification operation to a worst-case (most perceptible) contrast and frequency, and to apply the operation only if we would not expect a feature with that contrast and frequency to be visible.

Conceptually, the VDS algorithm consists of examining each node in the tree and deciding whether to fold that node.[1] Folding a node can affect the corresponding region of the image in many possible ways. As the vertices and triangles supported by the node merge and shift, features in the image may shrink, stretch, or disappear completely. Because this is a 3-D model, shifting triangles that lie on a visual 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 [Lindstrom 99]. At this point, however, the requisite rendering and image processing appears far too expensive for dynamic simplification. Instead, we want a conservative worst-case bound on the changes in the image caused by folding the node. Our goal is to evaluate a hypothetical change *at least* as perceptible as any changes that folding actually incurs. For this hypothetical change, we consider the removal of a feature with a worst-case contrast and spatial frequency.

To determine the worst-case spatial frequency induced by folding the node, we observe that each fold operation affects the triangles supported by the folded node. For a given viewpoint, view direction, and field of view, these triangles cover a certain region

[1] In practice, we need only traverse an *active boundary* that forms a cut across the vertex tree [XXX Luebke 97].

of the image on the screen. Folding the node will change this region, and only this region, in the image. The maximum screen-space extent of the region therefore determines the *minimum frequency* in the image that folding the node can affect. To a first approximation, features at lower frequencies are perceptible at lower contrast than high-frequency features, so we can use this minimum frequency as our worst-case bound.

The worst-case contrast of the fold operation is the maximum contrast between an image of the region of support at full resolution and an image of the simplified region after the fold. There are two basic cases:

- The entire region of support 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 intensities and relative positions of the vertices before and after the fold.
- The region of support includes a silhouette edge. This expands the possible contrast incurred by the fold operation to include the portion of the scene *behind* the region of support, since shifting a triangle may expose a very bright or very dark feature occluded before the fold.

Section 2 describes the details of our method for calculating the worst-case contrast and spatial frequency of a fold operation, and whether a node might contain a silhouette edge.

3.4 Simplifying assumptions

To make tractable the complex and computationally demanding task of applying perceptual metrics to polygonal simplification, we make several simplifying assumptions. First, we assume that our models have been pre-lit, with colors from the lighting calculation assigned to each vertex; the models in our experiments used either radiosity or Phong lighting. We base our decisions about contrast on vertex intensities. Though we could conceptually work from surface normals and consider contrasts based on dynamically changing lighting, the extra processing involved seems prohibitive. Techniques for reducing that processing cost may be a promising avenue for future work.

Our method of estimating contrast for front-facing regions also assumes a well-behaved simplification, which does not cause surface intersections or self-intersections. For example, consider two concentric cylinders, one bright and one dark. A simplification operation on the outer cylinder could easily introduce an intersection, which would induce contrast far greater than our technique would predict. A simplification strategy careful to prevent surface intersections, such as the *simplification envelopes* of [Cohen 96] could be used to ensure that this assumption holds, but our current simplification algorithm does not take such care. Note, however, that preventing surface intersections does not necessarily preclude topology-modifying operations such as closing holes or merging objects, and our current system permits these operations.

A final assumption bears particular mention. We use the psychophysical studies on contrast sensitivity and visual acuity described above to guide our decision on the perceptibility of a simplification operation. However, these studies measured *static contrast sensitivity*: the ability of the eye to perceive static stimuli at various contrasts, frequencies, eccentricities, and velocities. In applying their results to guide our dynamic simplification operations, we ignore a small but important temporal factor. Folding a node causes a change in the rendered image, and the human visual system is sensitive to sudden changes. Even if the difference between the original and simplified image is not perceptible, a sudden transition between the images may be visible as a small flicker. Our current system, based upon a static perceptual model, might not avoid such flickers. We discuss several possible approaches to addressing this temporal factor in Section 6.4.

4 DETAILS OF THE ALGORITHM

4.1 Spatial Frequency: Estimating Node Extent

As described above, the studies of [Campbell 68] indicate that an image feature will be imperceptible if none of its component frequencies is perceptible. The first step in determining whether a node may be imperceptibly folded, then, is to estimate a worst-case spatial frequency induced by the change. Lower frequencies tend to be more perceptible than high frequencies, down to a cutoff frequency—around two cycles per degree of arc—that varies according to contrast, eccentricity, etc. Fortunately, this cutoff falls below the frequencies of principal interest to us. Consequently, we can compute the worst-case spatial frequency induced in the image by folding a node by estimating the minimum frequency induced, and clamping that minimum to the cutoff frequency for the given contrast.

The minimum frequency induced by a simplification operation is bounded by the spatial extent of the resulting change in the image. The minimum frequency component of a region in the

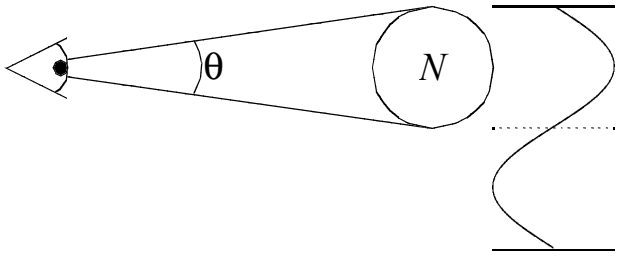

Figure 2: The minimum spatial frequency that can be affected by a node spanning θ° has one cycle per 2 θ°.

image spanning *n* degrees of the user's angular field of view is one cycle per 2*n* 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]. Since any change in the image caused by folding a node will occur within this region, the problem of computing the minimum frequency induced by the operation reduces to computing the screen-space extent of all triangles supported by the node.

Computing the screen-space extent, under perspective projection, of a portion of the model is a common operation in level-of-detail algorithms. Since exact extents are generally expensive to calculate, a standard approach is to use a simple bounding volume, such as a sphere or axis-aligned box, which contains the portion whose extent is to be calculated. The easily calculated screen-space extent of the bounding volume is then used as a conservative overestimate of the actual extent of the object or node under consideration. Our algorithm uses bounding spheres,

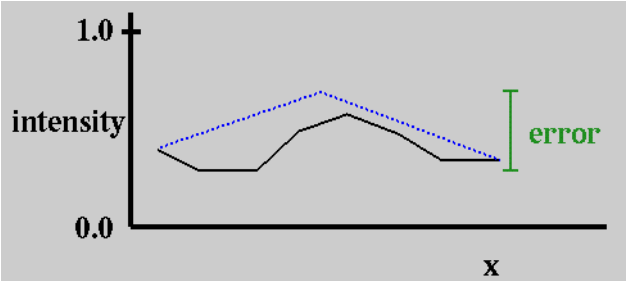

Figure 3: We use a conservative approximation of the maximum error in intensity induced by folding a node.

associating with each node a tight-fitting sphere that contains all triangles in the node's region of support. Bounding spheres are quite efficient; their angular extent as seen from a given viewpoint can be calculated very quickly [Luebke 97]. The minimum frequency affected by folding a node, then, is one cycle per degree of angular extent spanned by the node's bounding sphere.

4.2 Contrast: Estimating Intensity Change

Given a worst-case spatial frequency for the change induced by folding a node, we next need to compute the worst-case contrast of that change. Specifically, we wish to bound the maximum change in intensity between an image of the original model in the region supported by the node, and an image of the region after folding. For efficiency, we precompute this change in intensity and store it directly in the node structure. Here *intensity* is defined as the luminance output by a standard computer monitor, normalized to the range [0, 1].

As mentioned above, contrast induced by folding a node depends on whether the affected region lies on a model silhouette. Even in the simpler case of a front-facing interior region, determining the maximum contrast is an expensive process. Instead, we obtain a very conservative lower bound on that contrast 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 [Figure 3]. 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 conservative overestimates the contrast induced by folding a node, but appears to works sufficiently well in practice. Section 6.3 discusses possible improvements to the contrast calculation.

When the node supports a silhouette edge, we must be even more conservative. In the absence of knowledge about what lies behind the model, we must assume the worst: moving a silhouette edge might expose the darkest or brightest object in the scene, or the background color itself. Here we must compare the range of vertex intensities of the node's region of support against the brightest and darkest intensities in the scene, and treat the maximum possible difference in intensity as the contrast induced by the fold. Consequently, silhouette regions of the object are simplified less aggressively—exactly the behavior we should expect in a perceptually driven simplification algorithm.

4.3 Determining Silhouette Nodes

Since nodes affecting silhouette edges must be treated differently, we require an accurate and efficient method for identifying such nodes. For a given view, we define the *silhouette node*s as those nodes supporting both front-facing and back-facing triangles in the original mesh. Our initial technique for determining silhouette nodes used the *cone of normals* approach [Shirman 93] used by both [Luebke 97] and [Hoppe 97]. This approach computes a cone in the space of normals that contains the normals of all supported triangles. This cone, along with the viewpoint and node bounding volume, can be used to decide whether any normal on the surface might be on the silhouette.

Unfortunately, the cone of normals proved overly conservative for our needs; too many interior nodes were being classified as silhouette nodes. Instead, we used a novel approach based on the rapid backface culling technique of Zhang and Hoff [Zhang 97]. We map the Gauss sphere of normal space to a *normal cube* whose faces are tiled into cells; each cell represents all the normals that fall within that cell. In effect, we are quantizing the space of normals. Each node in the model stores a *normal mask*, a bit vector representing the normals of all its supported triangles. A bit in the mask is set if a triangle normal falls within the corresponding cell of the normal cube [Figure 4].

The accuracy of the normal mask is bounded only by the number of cells, which depends on the length of the bit vector. This improves significantly over the cone of normals, which can greatly overestimate the range of normals. The normal masks are efficient to compute, since they can be propagated up the vertex tree using bitwise-OR operations. Deciding whether the node might lie on the silhouette can also be made very efficient by precomputing two bitmasks. One represents the space of normals that might be backfacing, and the other the space of normals that might be frontfacing. A node may be on the silhouette if its normal mask overlaps with both the frontfacing and the backfacing bitmasks. The test to classify a silhouette node therefore amounts to two bitwise-AND operations, whose cost depends on the length of the bit vector. We chose 48 bytes (64 bits per face of the normal cube) as a good compromise between accuracy and storage requirements.

4.4 Computing Node Eccentricity

We define a node's eccentricity E as the angular distance from the fovea to the node's image on the retina. To calculate this, we find the angle between a vector from the viewpoint to the center of node and a *gaze vector* along the user's direction of gaze. To account for the node's extent, we subtract half the angle subtended by the bounding sphere of the node (note that we are already calculating this angle to determine the node's minimum spatial frequency). Finally, we subtract an error interval—currently one degree of arc—to account for limited accuracy of our eye tracker, and clamp the resulting angle of eccentricity at 0.

4.5 Putting It All Together

The perceptual findings summarized in Section 3.2 provide the equations necessary for determining the perceptibility of contrast gratings with increasing distance from the fovea. The cortical magnification factor M equates a spatial frequency α at the fovea to an equally perceptible spatial frequency β at a given eccentricity. M varies in across different regions of the retina; to

simplify our perceptibility test in a conservative manner we use the cortical magnification factor of the most sensitive *temporal region*. Combining all equations, we can solve for the threshold contrast at a given spatial frequency and eccentricity:

$$\alpha = (1 + 0.29\boldsymbol{E})\beta$$
$$\boldsymbol{c}_{threshold} = \delta\alpha^{-2}\boldsymbol{e}^{\alpha} \qquad \text{[Equation 5]}$$

If β is the measured spatial frequency, E the measured eccentricity, and $\boldsymbol{c}_{node}$ the measured contrast at a node in the vertex tree, then the node may be imperceptibly folded when $\boldsymbol{c}_{node} < \boldsymbol{c}_{threshold}$. Note that the value of $\boldsymbol{c}_{node}$ will depend on whether the node was found to lie on the silhouette.

4.6 Implementation Framework

We implemented our system in OpenGL on an SGI Onyx2 computer with InfiniteReality graphics. As mentioned above, we use the VDSlib package for view-dependent polygonal simplification. VDSlib allows users to plug in custom criteria for clustering, culling, simplifying, and rendering the vertex tree. The heart of our algorithm consists of two components. First, a preprocessing step augments an already-clustered vertex tree with data specific to our perceptual simplification process, such as the contrast induced by a fold operation and the normal mask used for silhouette detection. Second, a run-time callback examines nodes, using contrast, spatial frequency, and eccentricity to decide whether VDSlib should fold the node.

For eye tracking, we used ERICA™, a commercial off-the-shelf system developed by ERICA Inc. ERICA uses an infrared camera and a light-emitting diode (housed in a small box under the monitor) to illuminate the user's eye and track the resulting reflections. A standard PC analyzes images from the camera to extract the user's gaze direction, which was then sent to our graphics workstation via a serial connection. ERICA possesses several advantages for our purposes: it is fast (60 Hz update), non-intrusive, and accurate to half a degree of arc [Hutchinson 89]. Disadvantages include a short calibration step, required before every use, and sensitivity to sources of ambient infrared radiation, such as sunlight or incandescent lamps. However, the chief disadvantage of ERICA for our purposes is the lack of head tracking in the system. Consequently, the user must remain relatively still during use or risk losing tracking accuracy. ERICA, Inc. is developing a head-tracked system, but for our current studies we decided to provide a chin rest to ensure that high accuracy was maintained throughout.

5 RESULTS

Figures 1 and 6 shows models simplified with our perceptually driven gaze-directed algorithm. Not surprisingly, the reductions in polygon count are somewhat modest compared to the reductions achieved by algorithms making no perceptual guarantees. Still, these results clearly show the potential of gaze-directed simplification to imperceptibly reduce scene complexity.

5.1 User Study: Evaluating Imperceptibility

We performed a user study to evaluate our system more formally, determining whether our algorithm can indeed produce a simplification imperceptible from the original model, evaluating imperceptibility across a range of values for δ (see Equation 5). The study tested whether subjects could perceive the difference

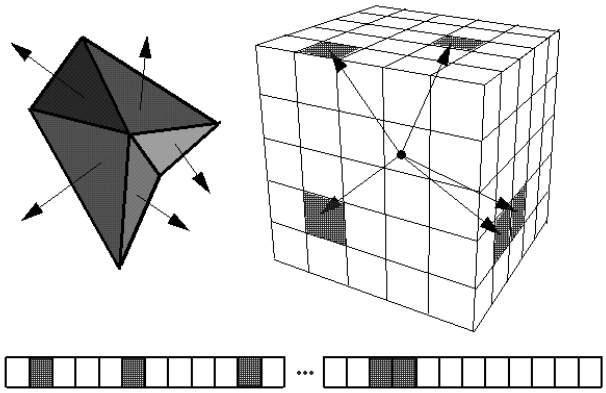

Figure 4: Efficient computation of silhouette nodes with the node's *normal mask*. Left: the node's supported triangles. Right: A cell in the normal mask is set if a normal falls within the corresponding range. Bottom: each cell corresponds to a bit in a bit vector.

between a rendering of a full-resolution model and a rendering of a model simplified with the gaze-directed algorithm described above. Since our current model of the perceptual system does not include temporal contrast sensitivity (see Section 3.4), we chose to eliminate this factor by fixing viewer gaze and avoiding sudden transitions between images.

The study consisted of eight subjects, each of whom performed 480 trials. During each trial, the subject fixated on a target (a short line segment) in the center of the screen. When the subject's gaze was fixed, they were shown two scenes in succession. The two scenes consisted of a single 3-D object and were identical in all parameters except resolution. For half the trials both images were presented at full resolution. For the remaining trials, one was presented at full resolution and the other was presented at reduced resolution using our gaze-directed simplification algorithm.

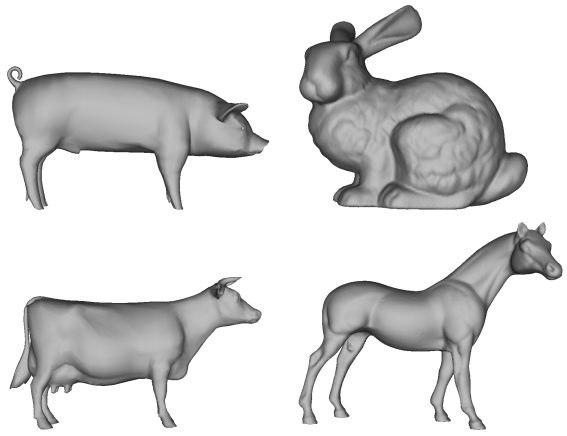

Figure 5: The four test models used in the user study: Pig (7040 polygons), Bunny (69,591 polygons), Cow (5804 polygons), Horse (96,966).

Each scene was displayed for 1000 milliseconds and the two scenes were separated for 500 milliseconds. A neutral grey background was displayed before, after, and between scenes. If

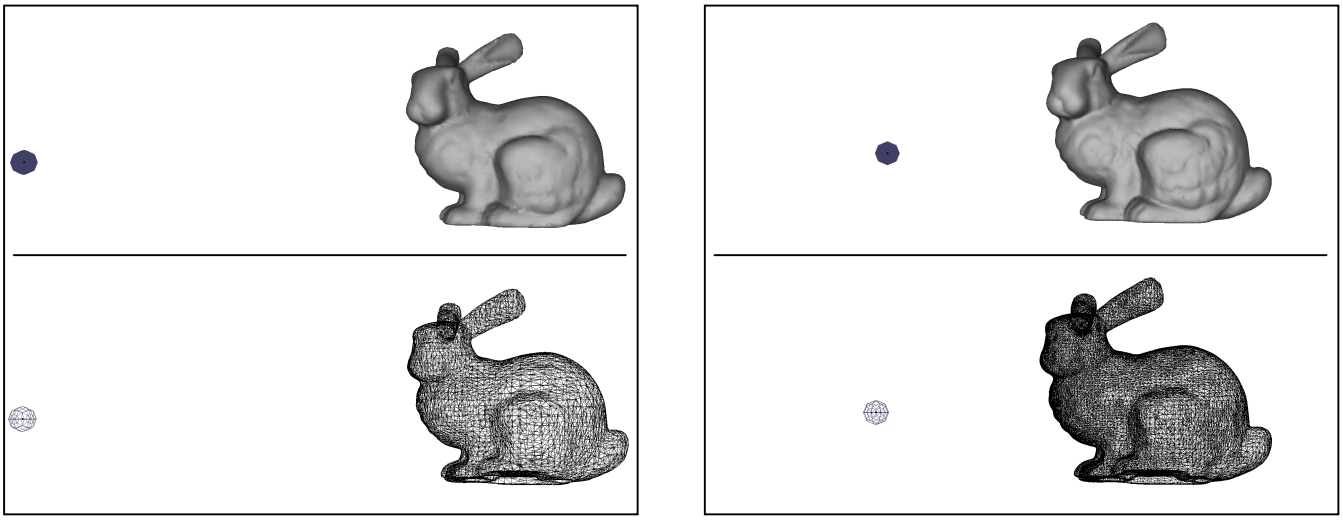

Figure 6: The Stanford Bunny model, rendered from the same viewpoint with different gaze points. The original model contains 69,591 polygons; the simplifications shown here contain 34,321 polygons (right) and 11,726 polygons (left). At a 46° field of view, the simplifications as shown are imperceptible.

the subject's gaze deviated by more than 2.3° from the center of the screen, the scene cleared immediately to the neutral background and the trial was cancelled. When the second scene finished displaying, the subject pressed Y or N on a keyboard to indicate whether they could detect any difference between the two images. To avoid subject fatigue, the next trial did not begin until 500 milliseconds after the subject had pressed a key.

After a practice session of 48 trials, each subject performed 480 trials in a continuous session. Subjects viewed four models (bunny, cow, horse, pig) from 6 random viewpoints for each threshold value δ used by the simplification algorithm. Viewing parameters were chosen so that the subject viewed the object at randomly distributed orientations from randomly distributed directions. Distance to the object was randomly chosen such that the visual angle subtended by the object was uniformly distributed from 5° to 60°. The screen occupied approximately 46° of the subject's field of view, so that in some views the object filled most of the screen. The value of δ was chosen from 10 values. Using a pilot study, we picked these values to span the range between slightly but clearly perceptible simplification and completely imperceptible simplification.

Eight subjects participated in the study. Each subject reported normal eyesight, some with corrective lenses. Subject accuracy is plotted against δ in Figure 6. *Baseline* represents the willingness of subjects to report a difference between the models when none existed. As the graph shows, subject accuracy deviates at lower δ values from baseline for models with substantial high spatial frequency content (bunny and horse). This confirms our prediction—at the correct scaling factor, simplification is imperceptible from the original model.

6 DISCUSSION AND FUTURE WORK

Our system shows the feasibility and potential of imperceptible simplification using gaze-directed rendering, but many avenues for further research remain. Below we discuss our results and address what we see as the most pressing and interesting directions for future work.

6.1 Results

We have demonstrated a novel approach to geometric simplification that is directly driven by perceptual rather than geometric criteria. Our approach simplifies the underlying polygonal model, imperceptibly reducing the number of polygons that must be transformed. We use a view-dependent simplification algorithm, enabling objects in the scene to span multiple levels of detail. Tracking the user's gaze with a commercial eye-tracking system allows us to obtain further simplification by reducing fidelity in the viewer's peripheral vision.

The simplification rates shown in Figures 1 and 6 may seem relatively modest by the standards of modern algorithms. This is not surprising, since we are making a guarantee on the fidelity of the resulting image that other algorithms do not make. To claim that the typical observer will be unable to perceive degradation, we make conservative choices throughout the algorithm. It is true, however, that many of our choices are overly conservative; the next section addresses some elements of the algorithm that

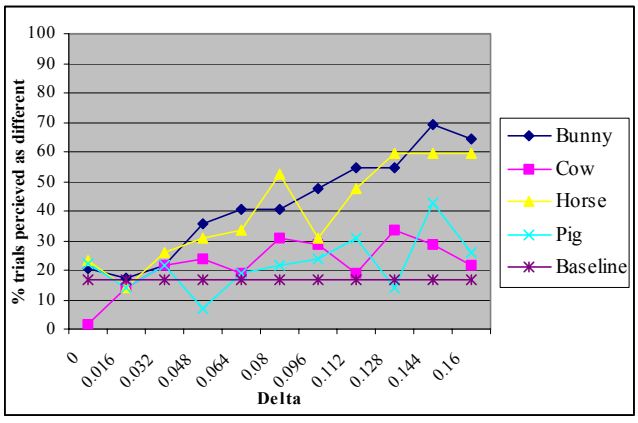

Figure 7: Subject accuracy versus δ (see Equation 5).

could be made more accurate to enable better simplification rates.

The user study verified that we are able to achieve imperceptible simplification. Under the conditions of the study, which include the ambient luminance of the test environment and the brightness and contrast of our CRT monitor, users were unable to distinguish simplified from original models better than a baseline performance at $\delta = 0.4$.

6.2 Applicability of Gaze-Directed Rendering

Gaze-directed rendering is a powerful concept with some clear limitations. First, correctly monitoring the user's gaze requires tracking the eye. Many methods for doing this are intrusive, requiring physical contact with the user's face or eyeball. Current non-intrusive techniques suffer their own disadvantages. For example, the ERICA system restricts the user's head to a small volume, requires a room without sunlight or incandescent lights, and involves a short calibration step before use.

It seems likely that eye-tracking technology will improve, eliminating these limitations. Some applications, however, are inherently difficult or impossible to improve with gaze-directed rendering. For example, if the display occupies a small field of view (under, say, 15°) the potential model degradation and performance increase due to eccentricity drops rapidly. On the other hand, for large field of view displays such as wide-angle head-mounted displays or CAVE™ immersive displays, gaze-directed simplification becomes very attractive, with the potential to reduce drastically the rendering workload. In fact, with such multi-screen wide-angle displays, head tracking again becomes a viable option.

Multiple viewers present an obvious problem for gaze-directed rendering, since viewers might examine different parts of the display at once. Such a scenario clearly increases the demand on the eye-tracking system and limits the degree of simplification possible. However, many rendering applications involve only a single user. As large and immersive displays grow more common, gaze-directed simplification offers an intriguing and powerful technique for managing rendering complexity.

6.3 Improving the Current System

We see many opportunities to improve the current system. The technique described in Section 4.2 for estimating the contrast induced by folding a node is far too conservative in most cases. We currently find the maximum possible difference in intensity between any pair of points in the node's region of support before and after simplification. However, carefully assigning coordinates and intensity of the node's proxy vertex can vastly improve on the actual maximum difference in intensity between the surfaces [Figure 7]. Consequently, the contrast c_{node} that we compare to the threshold contrast $c_{threhsold}$ is often far larger than necessary, forcing us to leave unfolded a node that could have been folded. We suspect that a technique that kept a tighter bound on the actual contrast induced by folding nodes would halve the number of polygons used in many simplifications.

The construction of the vertex tree also offers much room for improvement. We use the default clustering provided by VDSlib, based on the *tight octree* method of [Luebke 97]. This technique is simple, fast, and robust, but can result in clusterings far from optimal for our purposes. In particular no attention is paid to minimizing the intensity difference of the clustered vertices, which drives the contrast induced by folding those vertices. Constructing the hierarchy with more careful attention to the contrast and spatial extent spanned by the resulting nodes would certainly improve our results.

Tighter bounding volumes would also aid in calculating the spatial extent of the node. We currently use spheres to bound each node's region of support, but this can significantly overestimate the size of the region. Using tighter bounding volumes, such as oriented bounding boxes or oriented ellipsoids, would improve our estimates of spatial frequency, which in turn would improve the amount of simplification possible at a given contrast.

Many additional user studies on perceptually based simplification could be performed. It is important to validate more formally our claim of a model capable of imperceptible simplification. Testing across a greater range of subjects, models, and viewing conditions would help support that claim. More tests to evaluate the effect of gaze-directed simplification on task performance would also be valuable.

6.4 Extending the Perceptual Model

More fundamental improvements will require extending the perceptual model underlying simplification. Our most pressing need is for a perceptual model that accounts for temporal sensitivity. The human visual system is sensitive to flicker, so abrupt changes in the rendered image caused by folding a node may be perceptible when a gradual transition would not be. This sensitivity to flicker does not decrease with eccentricity in the same fashion as threshold contrast, so simply scaling our current equations is not sufficient. A more sophisticated perceptual model that accounted for *temporal contrast sensitivity* could prevent folds that would cause a visible "pop". Another possibility would be to soften the transition using alpha blending or *geomorphs* [Hoppe 97]. Here the perceptual model would indicate how many frames the folding transition should span to eliminate visible flicker.

Incorporating dynamic lighting into the visual contrast is another obvious avenue for future research. Lighting calculations such as the Phong model depend on the surface normal. One approach might be to use the current normal masks, which bound the space of normals subtended by each node's region of support, to compute the minimum and maximum intensities of that region

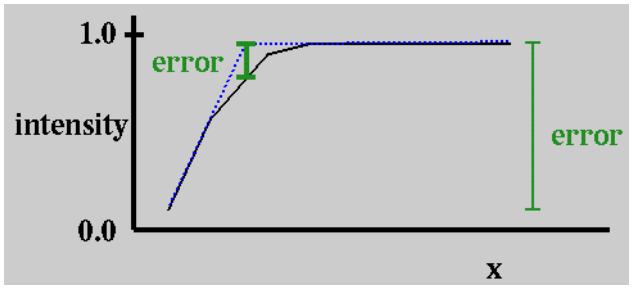

Figure 6: Overly conservative contrast estimation. The green bar on the left represents the actual maximum difference in intensity before and after folding the node; the bar on the right represents our estimated error.

under the given lighting conditions. The non-linearity of the Phong model complicates the situation, but a solution certainly appears feasible.

As mentioned above, our current method for estimating the contrast induced by a fold is overly conservative. One intriguing approach to addressing this problem would be to integrate the texture-based approach of Cohen *et al* [Cohen 98]. Their technique transforms vertex coloration to a texture map parameterized across the surface being simplified, and tracks the maximum texture map distortion introduced during simplification. If we could use this maximum distortion to bound the maximum intensity change introduced by a single fold operation, we might be able to achieve significantly more accurate estimates of contrast.

Many more user studies on perceptually based simplification could be performed. It is important to validate more formally our claim of a model capable of imperceptible simplification. Testing across a greater range of subjects, models, and viewing conditions would help support that claim. More tests to evaluate the effect of gaze-directed simplification on task performance would also be valuable. We are current designing and conducting two such tests, evaluating user performance with and without gaze-directed simplification on a search task and a naming-times task.

7 REFERENCES

[Campbell 66] Campbell, F.W. and Gubisch, R.W. Optical Quality of the Human Eye, *Journal of Pysiology*, 186 (1966)

[Campbell 68] Campbell, F.W. and Robson, J.G. An Application of Fourier Analysis to the Visibility of Contrast Gratings, *Journal of Physiology*, 187 (1968)

[Clark 76] Clark, James H. "Hierarchical Geometric Models for Visible Surface Algorithms," *Communications of the ACM*, Vol 19, No 10, pp 547-554.

[Cohen 96] Cohen, J., A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, W. Wright. "Simplification Envelopes", *Computer Graphics*, Vol 30 (SIGGRAPH 96).

[Cohen 98] Cohen, Jon, M. Olano, and D. Manocha. "Appearance-Preserving Simplification," *Computer Graphics*, Vol. 32 (SIGGRAPH 98).

[Cosman 81] Cosman, M., and R. Schumacker. "System Strategies to Optimize CIG Image Content". *Proceedings Image II Conference* (Scotsdale, Arizona), 1981.

[Funkhouser 93] Funkhouser, Tom, and C. Sequin. "Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments", *Computer Graphics*, Vol. 27 (SIGGRAPH 93).

[Garland 97] Garland, Michael, and P. Heckbert, "Survey of Polgyonal Surface Simplification Algorithms", SIGGRAPH 97 course notes (1997).

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

[Hutchinson 89] Hutchinson, Thomas E. "Human-Computer Interaction Using Eye-Gaze Input", *IEEE Transactions on Systems, Man, and Cybernetics* Vol 19, No. 6 (Novermber 1989).

[Kelly 75] Kelly, D.H. Spatial Frequency Selectivity in the Retina, *Vision Research*, 15 (1975)

[Koenderink 78] Koenderink, J.J. *et al.* Perimetry of contrast detection thresholds of moving spatial sine wave patterns. III. The target extent as a sensitivity controlling parameter. *Journal of the Optical Society of America*, 68 (1978)

[Lindstrom 99] Lindstrom, P. and Turk, G. To appear in ACM Transactions on Graphics. Available as technical report GIT-GVU-99-49. December 1999.

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

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

[Reddy 97] Reddy, Martin. "Perceptually-Modulated Level of Detail for Virtual Environments", Ph.D. thesis, University of Edinburgh, 1997.

[Rovamo 79] Rovamo, J. and Virsu, V. An Estimation and Application of the Human Cortical Magnification Factor, *Experimental Brain Research*, 37 (1979)

[Savoy 75] Savoy, R.L. and McCann, J.J. Visibility of low-spatial-frequency sine-wave targets: Dependence on number of cycles, *Journal of the Optical Society of America*, 65 (1975)

[Shirman 93] Shirman, L., and Abi-Ezzi, S. "The Cone of Normals Technique for Fast Processing of Curved Patches", Computer Graphics Forum (*Proc. Eurographics '93*) Vol 12, No 3, (1993), pp 261-272.

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

[Zhang 97] Fast Backface Culling Using Normal Masks , Hansong Zhang, Kenny Hoff, ACM Symposium on Interactive 3D Graphics, Providence, 1997.