

Gaze-Contingent Level Of Detail Rendering

Hunter Murphy[†] and Andrew T. Duchowski[†]

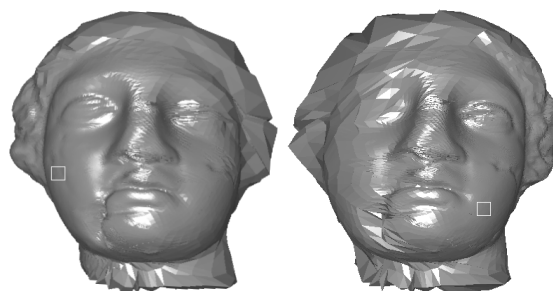
Department of Computer Science, Clemson University, Clemson, SC, USA

Abstract

The contributions of this paper are the development and evaluation of a nonisotropic model-based Level Of Detail (LOD) rendering technique for gaze-contingent viewing of multiresolution meshes. A high resolution portion of the model is rendered at the eye-tracked Point Of Regard (POR). A method is given for converting a closed polygonal mesh to a nonisotropic LOD representation suitable for gaze-contingent viewing. Based on a theoretical model of visual acuity, a three-dimensional spatial degradation function is obtained from human subject experiments in an attempt to render imperceptibly degraded geometric objects. Unlike previous LOD approaches, our resolution degradation method is based on the measurement of visual angle in world coordinates and is applied directly to object geometries prior to rendering. The gaze-contingent technique is evaluated in a Virtual Reality (VR) system integrated with a binocular eye tracker. To our knowledge, this is the first example of a binocular eye-tracked VR system used to evaluate a gaze-contingent modeling technique. Results are reported in terms of rendering performance, indicating an overall 4-fold average frame rate improvement during gaze-contingent viewing. Frame rate improvement ranged from a factor of at least 2, up to a 15-fold gain in performance over full resolution display, varying with the model complexity and the instantaneous direction of the viewer's gaze.

1. Introduction

To increase display rates above those currently provided by view-dependent Level Of Detail (LOD) rendering methods, it has been suggested that an eye tracker is required to enable the presentation of high resolution portions of the scene or object only at the point of highest visual acuity, i.e., at the foveal Region Of Interest (ROI).²⁰ The motivation behind such proposed *gaze-contingent* systems is to minimize overall display bandwidth requirements by reducing peripheral information in concordance with the perceptual limits of the Human Visual System (HVS). Efforts at providing peripherally degraded information date back to early eye-slaved flight simulators. Since then, development of similar screen-based peripheral degradation approaches has continued to show promise for gaze-contingent display acceleration with minimal cost to either perception or performance. Model-based approaches, however, have not enjoyed similar progress. Although a good deal of work has recently been devoted to the development of mesh subdivision techniques for multiresolution simplification of complex models,³² we are not aware of any published results concerning display



(a) POR over left cheek. (b) POR below right cheek.

Figure 1: Two frames during gaze-contingent viewing of igea model with superimposed POR boxes.

speedup showing successful adaptation of these techniques within a true gaze-contingent system, i.e., one where an eye tracker is employed.

The contributions of this paper are the development and evaluation of a nonisotropic model-based LOD rendering technique for gaze-contingent viewing of multiresolution meshes, where a high resolution portion of the model is rendered at the eye-tracked Point Of Regard (POR), as shown in Figure 1. Since mesh geometries are generally not stored

[†] {hmunphy | andrewd}@vr.clemson.edu

with the information necessary for such a display scheme, a method is given for converting closed polygonal meshes to representations suitable for gaze-contingent display. The modeling technique is then evaluated in a Virtual Reality (VR) system integrated with a binocular eye tracker. Evaluation consists of the determination of an empirical resolution degradation function and the measurement of resultant rendering performance.

The paper is organized as follows. Section 2 briefly surveys eye tracking applications and gaze-contingent interaction techniques, and summarizes previous geometric modeling work suitable for gaze-contingent display. Graphics and eye tracking hardware components are described in Section 3, along with techniques related to eye tracker and VR system integration provided in Section 4. Section 5 gives details of the nonisotropic LOD technique developed for gaze-contingent viewing in VR, and Section 6 gives the spatial acuity degradation function obtained from human subject trials. Section 7 reports the resultant rendering performance, followed by concluding remarks in Section 8.

2. Background

Interest in gaze-contingent interface techniques has endured since early implementations of eye-slaved flight simulators and has since permeated several domains including human-computer interaction, teleoperator environments, and visual communication modalities.⁸ The use of eye trackers in graphical systems falls into two general application types: *diagnostic* and *interactive*.

Diagnostic applications typically involve the recording of eye movements over time, i.e. scanpaths, for *post facto* analysis of the user's overt visual attention over a given stimulus. For example, Duchowski et al. use an eye tracker to record eye movements in a virtual aircraft cargo bay for inspection training.⁴ Other examples of diagnostic applications can be found elsewhere.⁶ Diagnostic systems generally do not require the display to react to the user's gaze, and are therefore outside the scope of this paper.

Interactive eye tracking systems, on the other hand, typically respond in some way to the location of the user's gaze. Early noteworthy examples date back to the work of Starker and Bolt²⁷ and Jacob.¹¹ Such interactive systems may be classified by two application sub-types: *selective* and *gaze-contingent*. The latter can be further delineated in terms of display processing, as shown in the hierarchy in Figure 2.

In selective applications, as exemplified by Jacob's work, the user's gaze acts as an alternate mode of input, often compared to a pointing device. For example, Tanriverdi and Jacob have recently used an eye tracker as a selection device in VR.²⁸

In gaze-contingent displays, the objective is to partition the display into two imperceptible spatial regions, a high-

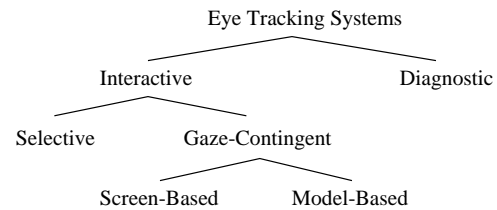


Figure 2: Hierarchy of eye tracking applications.

resolution foveal ROI surrounded by a low-resolution peripheral region. There are two main approaches: *screen-based* and *model-based*. The former deals with the manipulation of framebuffer contents just prior to display. The periphery of the display is often masked or smoothed in some way, reducing the bandwidth requirements by compressing the information (in bits-per-pixel) required to display or transmit the final image.

The idea of gaze-contingent displays is not new and dates back to early military applications.^{13,17} In the Super Cockpit Visual World Subsystem, Kocian considered visual factors including contrast, resolution, and color in the design of a head-tracked display. In their Simulator Complexity Testbed (SCTB), Longridge et al. included an eye-slaved ROI as a major component of the Helmet Mounted Fiber Optic Display (HMFOD). This ROI provided a high resolution inset in a low resolution (presumably homogeneous) field which followed the user's gaze. The precise method of peripheral degradation was not described apart from the criteria of low resolution. However, the authors did point out that a smooth transition between the ROI and background was necessary in order to circumvent the possibility of a perceptually disruptive edge artifact.

Recently, sophisticated approaches have been developed for ROI-based image and video coding.^{14,5,18,23} For screen-based VR rendering the work of Watson et al. is particularly relevant.³¹ The authors studied the effects of LOD peripheral degradation on visual search performance. Both spatial and chrominance detail degradation effects were evaluated in Head Mounted Displays (HMDs). To sustain acceptable frame rates, two polygons were texture mapped in real-time to generate a high resolution inset within a low resolution display field. The authors suggested that visual spatial and chrominance complexity can be reduced by almost half without degrading performance.

In an approach similar to Watson's, Reddy used a view-dependent screen-based LOD technique to evaluate both perceptual effects and system performance gains.²⁴ The author reported a perceptually modulated LOD system which affords a factor 4.5 improvement in frame rate. It is not entirely clear how the LOD model was constructed, i.e. what was the method of degradation, nor is it clear what kind of apparatus was used. Reddy's empirical evaluation of the LOD model was performed on a 43.6×33.4 degree Field Of View (FOV) display, presumably a desktop monitor without the use of an eye tracker.

As an alternative to the screen-based peripheral degradation approach, model-based methods aim at reducing resolution by directly manipulating the model geometry prior to rendering. The technique of simplifying the resolution of geometric objects as they recede from the viewer, as originally proposed by Clarke,¹ is now standard practice, particularly in real-time applications such as VR.²⁹ Clarke's original criteria of using the projected area covered by the object for descending the object's LOD hierarchy is still widely used today. However, as Clarke suggested, the LOD management typically employed by these polygonal simplification schemes relies on pre-computed fine-to-coarse hierarchies of an object. This leads to uniform, or *isotropic* in terms of resolution degradation, simplification of objects.

A gaze-contingent model-based adaptive rendering scheme was proposed by Ohshima et al., where three visual characteristics were considered: central/peripheral vision, kinetic vision, and fusional vision.²² The LOD algorithm generated isotropically degraded objects at different visual angles. Although the use of a binocular eye tracker was proposed, the system as discussed used only head tracking as a substitute for gaze tracking.

Isotropic object degradation is not always desirable, especially when viewing large objects at close distances. In this case, traditional LOD schemes will display an LOD mesh at its full resolution even though the mesh may cover the entire field of view. Since acute resolvability of human vision is limited to the foveal 5°, object resolution need not be uniform. This is the central tenet of gaze-contingent systems.

Numerous multiresolution mesh modeling techniques suitable for gaze-contingent viewing have recently been developed.³² Techniques range from multiresolution representation of arbitrary meshes to the management of LOD through peripheral degradation within an HMD where gaze position is assumed to coincide with head direction.^{16, 21, 9, 33, 25} Although some of these authors address view and gaze dependent object representation, few results concerning display speedup are as yet available showing successful adaptation of these techniques within a true gaze-contingent system, i.e. one where an eye tracker is employed. Due to the advancements of multiresolution modeling techniques and to the increased affordability of eye trackers, it is now becoming feasible to extend the LOD approach to gaze-contingent displays, where models are rendered *non-isotropically*.

An early example of a nonisotropical model-based gaze-contingent system, where gaze direction is directly applied to the rendering algorithm, was presented by Levoy and Whitaker.¹⁵ The authors' spatially adaptive near real-time ray tracer for volume data displayed an eye-slaved ROI by modulating both the number of rays cast per unit area on the image plane and the number of samples drawn per unit length along each ray as a function of local retinal acuity. The ray-traced image was sampled by a nonisotropic con-

volution filter to generate a 12° foveal ROI within a 20° mid-resolution transitional region. Based on preliminary estimates, the authors suggested a reduction in image generation time by a factor of up to 5. An NAC Eye Mark eye tracker was used to determine the user's POR while viewing a conventional 19" TV monitor. A chin rest and immobilization strap were used to eliminate the need for head tracking.

Danforth et al. used an eye tracker as an indicator of gaze in a gaze-contingent multiresolution terrain navigation environment.² A surface, represented as a quadrilateral mesh, was divided into fixed-size (number of vertices) sub-blocks, allowing rendering for variable LOD on a per-sub-block basis. Resolution level was chosen per sub-block, based on viewer distance. The resolution level was not discrete; it was interpolated between the pre-computed discrete levels to avoid "popping" effects. The approach used is reasonably effective; however, it is not clear whether the technique is applicable to arbitrary meshes.

Perhaps the most closely related work to the current modeling approach is that of Luebke et al. Recently, the authors have developed a gaze-directed LOD technique to facilitate the gaze-contingent display of geometric objects.¹⁹ To test their rendering approach the authors employed a table-mounted monocular eye tracker to measure the viewer's real-time location of gaze over a desktop display. While this work shows the feasibility of employing an eye tracker, the implementation framework used by the authors lacked a head tracker and required a chin rest to ensure tracker accuracy.

Here we present an object-based LOD method, similar to that of Ohshima et al., where objects are modeled for gaze-contingent viewing. Unlike their approach, resolution degradation is applied nonisotropically, i.e. objects are not necessarily degraded uniformly. We chose to follow the work of Eck et al.⁷ as a suitable starting point for developing our modeling technique. We summarize the main points of our strategy, identifying differences of our implementation where appropriate. We then discuss a spatial degradation function obtained from human subject trials.

It should be noted that our spatial degradation function for LOD selection differs significantly from the area-based criteria originally proposed by Clarke. Instead of evaluating the screen coverage of the projected object, our degradation function is based on the evaluation of visual angle in world coordinates. Since gaze-contingent LOD management relies on the selection of object polygons for multiresolution reconstruction it was deemed simpler to follow a ray casting approach, as suggested by Levoy and Whitaker,¹⁵ rather than to pursue a polygonal screen coverage calculation as proposed by Clarke.

System performance measurements are obtained from experiments using a binocular eye tracker built into an HMD. To our knowledge, this is the first example of a binocular eye-tracked Virtual Reality system used to evaluate a gaze-contingent modeling technique.

3. Hardware Components

Our primary rendering engine, housed in the Virtual Reality Eye Tracking (VRET) lab at Clemson, is a dual-rack, dual-pipe, SGI Onyx2® InfiniteReality2™ system with 8 raster managers and 8 MIPS® R12000™ processors, each with 8MB secondary cache. It is equipped with 8Gb of main memory and 0.5Gb of texture memory.

Multi-modal hardware components include a binocular eye tracker mounted within a Virtual Research V8 (high-resolution) HMD. The V8 HMD offers 640×480 resolution per eye with individual left and right eye feeds. HMD position and orientation tracking is provided by an Ascension 6 Degree-Of-Freedom (6DOF) Flock Of Birds (FOB).

The eye tracker is a video-based, corneal reflection unit, built jointly by Virtual Research and ISCAN. Each of the binocular video eye trackers is composed of a miniature digital camera and infrared light sources, with the dual optics assemblies connected to a dedicated personal computer (PC). The ISCAN RK-726PCI High Resolution Pupil/Corneal Reflection Processor uses corneal reflections (first Purkinje images) of infra-red LEDs mounted within the helmet to measure eye movements. Figure 3 shows the dual cameras and infra-red LEDs of the binocular assembly, with a picture of the helmet inset at top-center. The processor operates

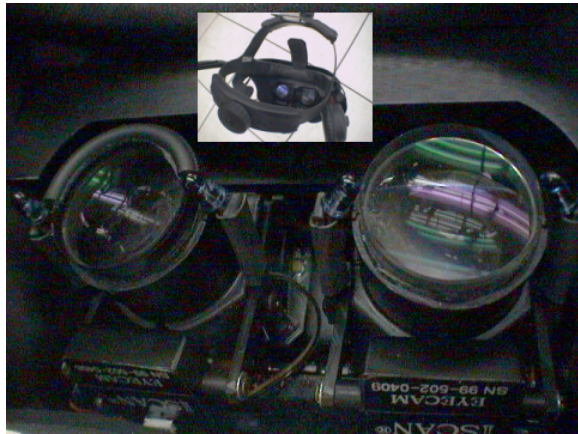


Figure 3: Closeup of HMD binocular eye tracker optics.

at a sample rate of 60Hz and the subject's eye position is determined with an accuracy of approximately 0.3 degrees over a ± 20 degree horizontal and vertical range using the pupil/corneal reflection difference. The maximum spatial resolution of the calculated POR provided by the tracker is 512×512 pixels per eye.

The binocular eye tracking assembly allows the measurement of vergence eye movements, which in turn provides the capability of calculating the three-dimensional virtual coordinates of the viewer's gaze. Using the vendor's proprietary software and hardware, the PC calculates the subject's real-time POR from the video eye images. In the current

gaze-contingent VR configuration, the eye tracker is treated as a black box delivering real-time eye movement coordinates (x_l, y_l, t) and (x_r, y_r, t) over a 19.2 Kbaud RS-232 serial connection, and can be considered as an ordinary positional tracking device.

4. Eye Tracking

4.1. Eye Tracker Coordinate Mapping

A critical concern in designing a gaze-contingent VR system is the mapping of eye tracker coordinates to the application program's reference frame. The eye tracker calculates the viewer's POR relative to the eye tracker's screen reference frame, e.g., a 512×512 pixel plane, perpendicular to the optical axis. The eye tracker returns a sample POR coordinate pair for each eye. These coordinate pairs must be mapped to the extents of the application program's viewing window.

Raw eye tracker coordinates are in the range [0,511]. In practice, the usable, or effective, coordinates are dependent on: (a) the size of the application window, and (b) the position of the application window, both relative to the eye tracker's reference frame. Proper mapping between eye tracker and application coordinates is achieved through the measurement of the application window's extents in the eye tracker's reference frame. This is accomplished by using the fine cursor movement and cursor location readout of the eye tracker.

To obtain the extents of the application window in the eye tracker's reference frame, the application window's corners are measured with the eye tracker's cursor. Figure 4 illustrates an example of a 600×450 application window as it would appear on the eye tracker scene monitor. Given the ex-

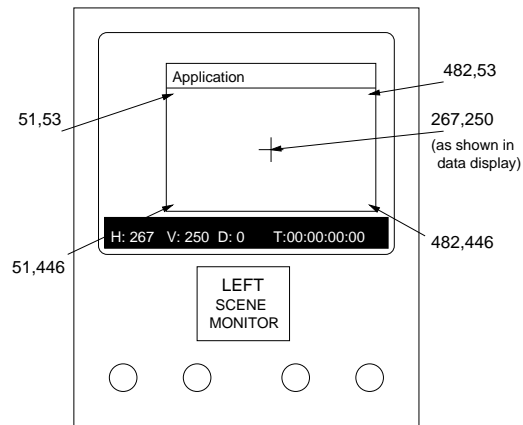


Figure 4: Mapping measurement example.

tents of both application and eye tracker screen coordinates, a simple linear interpolation mapping is used to map raw POR data to the graphics screen coordinates.⁴ While seemingly trivial, this mapping is key to proper calculation of the gaze vector in world coordinates from raw POR data and is

also essential for alignment of target points displayed by the application program during calibration of the eye tracker.

4.2. Gaze Vector Calculation

The calculation of gaze in three-space depends only on the relative positions of the two eyes on the horizontal axis. The parameters of interest are the three-dimensional virtual coordinates, (x_g, y_g, z_g) , which can be determined from traditional stereo geometry calculations.¹⁰ Helmet tracking determines both helmet position and the (orthogonal) directional and up vectors, which determine head-centric coordinates.

Given instantaneous eye tracked coordinates, (x_l, y_l) and (x_r, y_r) , in the left and right image planes (mapped from eye tracker screen coordinates to the near view plane), and head-tracked head position coordinates, (x_h, y_h, z_h) , the coordinates of the gaze point, (x_g, y_g, z_g) , are determined by the relations:

$$x_g = (1 - s)x_h + s(x_l + x_r)/2 \quad (1)$$

$$y_g = (1 - s)y_h + s(y_l + y_r)/2 \quad (2)$$

$$z_g = (1 - s)z_h + sf \quad (3)$$

where $s = b/(x_l - x_r + b)$, b is the baseline distance between the left and right eye centers, and f is the distance to the near viewing plane along the head-centric z -axis.

Note that since the vertical eye tracked coordinates y_l and y_r are expected to be equal (since gaze coordinates are assumed to be epipolar), the vertical coordinate of the central view vector defined by $(y_l + y_r)/2$ is somewhat extraneous; either y_l or y_r would do for the calculation of the gaze vector. However, since eye tracker data is also expected to be noisy, this averaging of the vertical coordinates enforces the epipolar assumption.

To enable gaze-contingent LOD rendering, it is necessary to calculate the intersection of the user's gaze with the environmental polygons. To calculate gaze direction, the gaze point is expressed parametrically as a point on a ray with origin (x_h, y_h, z_h) , with the ray emanating along a vector scaled by parameter s . That is, rewriting Equations (1)–(3) in vector notation, $\mathbf{g} = \mathbf{h} + s\mathbf{v}$, where \mathbf{h} is the head position, \mathbf{v} is the central gaze vector and s is the scale parameter as defined previously. To align the gaze vector with the current head orientation, it is first transformed to the instantaneous head-centric reference frame by multiplying the gaze vector \mathbf{v} by the orientation matrix returned by the head tracker. Further technical details of the eye tracking system integration may be found elsewhere.⁶

5. Multiresolution Geometric Modeling

The process of converting the original mesh to a multiresolution representation involves three primary steps: partitioning the mesh into tiles, mapping the tiles to n -gons, and remeshing into multiresolution form. These steps closely follow the

work of Eck et al.;⁷ only a brief review of this approach is given with deviations from the original process identified where appropriate.

5.1. Tiling

Because mesh files rarely contain explicit connectivity information, implicit connectivity information is extracted by growing tiles across the entire surface of the mesh. Tiles partition a mesh into local groups where each tile is a contiguous subset of the original mesh, i.e., a set of faces enclosed within a single non-self-intersecting boundary. Tiling is facilitated through the use of a Voronoi diagram, in this case a group of tiles in which the geodesic distance from any face in a tile to the centroid of the tile is shorter than its distance to any other tile's centroid. Since the mesh geometry is inherently discrete, a true Voronoi diagram is difficult to construct. A discrete approximation is used instead.

Construction of the Voronoi diagram is accomplished by using Dijkstra's Shortest Path algorithm. A priority queue is initialized with a random face that serves as the seed for a new tile. The face with the shortest geodesic distance from the centroid of its assigned tile is removed and checked against the existing faces in the tile for violations of the single boundary rule. A triangle violating this rule is added to the priority queue as a new tile seed. Assuming no violation occurred, the face's neighbors are added to the priority queue and the process is repeated until the priority queue is empty.

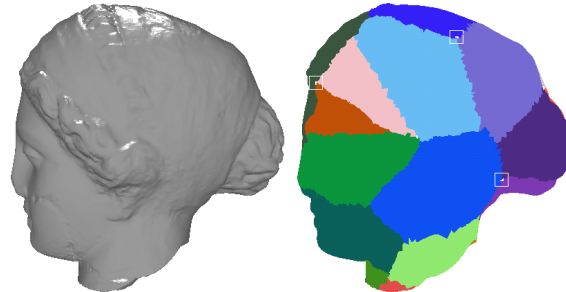


Figure 5: Original mesh and Voronoi tiling with boxes superimposed over selected corner vertices.

Upon completion of the initial partitioning process two additional conditions are addressed. First, exactly three tiles must meet at any corner (See Figure 5). Examination of the faces sharing a corner vertex can reveal violations, resulting in one of the faces being added to the priority queue as a new tile seed. Second, a tile may share only one edge with a neighbor. If more than two corners are shared between tiles, a violation occurs and a face from the offending corner vertex is added to the priority queue as a new tile seed. Enforcement of these two rules allows the Voronoi tiling to be converted to its dual Delaunay triangulation.

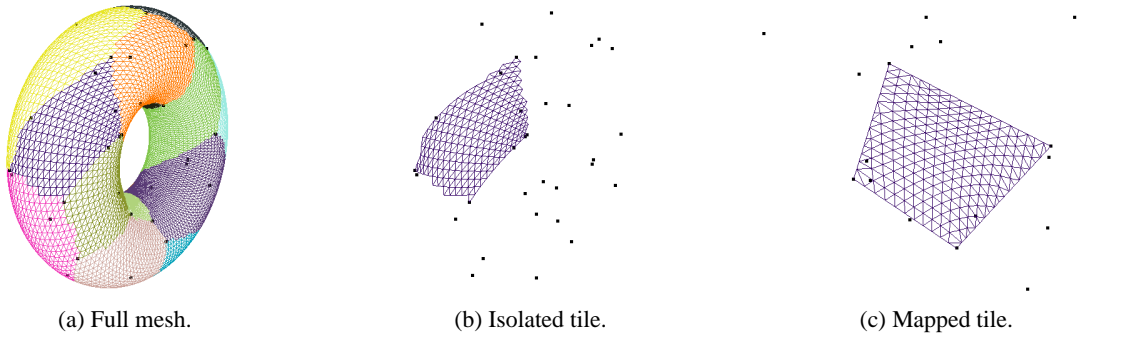


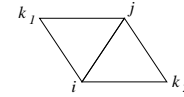
Figure 6: Harmonic mapping.

A final smoothing operation is applied to each tile to ensure that each boundary vertex is of degree three, which is required during the mapping process to prevent the formation of degenerate triangles. If a non-corner vertex with only two edges is discovered, the face in the tile to which the vertex belongs is surrendered to the tile with which the boundary is shared.

Eck et al. designed their tiling approach to provide subdivision connectivity, a requirement for their wavelet-based LOD models. Wavelet-based geometric reconstruction was deemed too computationally costly for real-time non-isotropic mesh reconstruction, obviating the need for Delaunay triangulation. In addition, only closed meshes were considered, further reducing the computational burden. If the tiles cover the entire mesh and adhere to the three requirements specified above, mapping of the tiles can occur. A completed tiling of the *igea* model is shown in Figure 5.

5.2. Harmonic Mapping

Mapping each tile to a planar n -gon can be visualized as pegging the corner vertices of an elastic tile and allowing it to flatten under its own internal tension. Piecewise linear harmonic maps, by minimizing total elastic energy, minimize distortion of the tile's topology. The n -gon to which the tile is pegged is constructed by affixing the corner vertices to a circle with arc-lengths between corner vertices proportional to tile edge lengths in the original mesh. These new coordinates are assigned to the appropriate corner vertices. Boundary vertices are also assigned new coordinates by maintaining the ratio between the original corner vertex positions and the new corner vertex positions. Spring constants $\kappa_{i,j}$, shown in Figure 7 as derived by Eck et al., are defined along triangle edges to minimize topological distortion. In some cases uniform spring constants must be used to allow the system to converge. While the total energy $\sum \kappa_{i,j} \|e_{i,j}\|$ is greater than a preset limit, each vertex is displaced by the scaled sum of forces acting directly on it. When the energy of the tile has been minimized, the original points have been mapped to locations on the n -gon. A mapped tile is shown in Figure 6.



$$\kappa_{i,j} = \left(\|e_{i,k_1}\|^2 + \|e_{j,k_1}\|^2 - \|e_{i,j}\|^2 \right) / \text{Area}_{i,j,k_1} + \left(\|e_{i,k_2}\|^2 + \|e_{j,k_2}\|^2 - \|e_{i,j}\|^2 \right) / \text{Area}_{i,j,k_2}$$

Figure 7: Spring constants used in harmonic mapping.

5.3. Remeshing

Once all the points in a tile are mapped, the new mesh can be projected onto the planar n -gon. The first of the multiresolution levels is constructed by connecting two adjacent corners of the map to the center of the map, creating one face. Each face is then 4-to-1 split by bisecting its edges (referred to as parametrically uniform resampling by Eck et al.) creating four new faces in the next level of subdivision. These are stored in a complete quadtree data structure for efficient rendering. The process continues recursively to a predefined level of subdivisions, each level a closer approximation to the original mesh than its predecessor. Once the new mesh has been defined on the n -gon, the 2D points are mapped to 3D from their barycentric coordinates defined for each face in the original mesh. This means that each point in the new mesh lies exactly on the surface of the original mesh. When this process has been repeated for every tile, a new multiresolution approximation of the original mesh has been created.

5.4. Geometric Reconstruction

Given a multiresolution representation of an object, it can now be rendered in a gaze-contingent manner. Our approach is to calculate the visual angle θ in world coordinates between the gaze vector \mathbf{v} and the vertices of each triangle in the level one (coarsest) geometry. An acuity-based resolution degradation function (see below) modeling peripheral falloff is used to select triangles for further subdivision (higher resolution approximation). If any vertex in the triangle being tested falls within the current range of the degradation function, each triangle in its next level of subdivision is recursively tested at the next highest degradation level. For

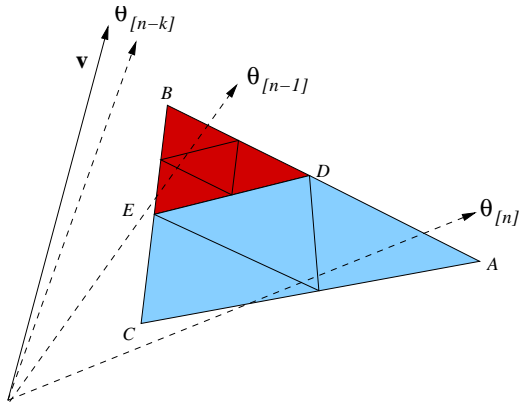


Figure 8: Gaze-contingent geometric reconstruction.

example, as illustrated in Figure 8, ΔABC is tested against the furthest range of the degradation function. Vertices B and C fall within that range, so ΔABC is tested at its next highest resolution (subdivision level), this time against $\theta_{[n-1]}$. Vertex B of ΔDBE falls within that range, resulting in ΔDBE being tested at its next highest resolution against $\theta_{[n-2]}$.

6. Estimating Resolution Degradation

An experiment was conducted to find a resolution degradation function and to compare the resultant function with a theoretical model of visual acuity. At photopic light levels (daylight) foveal acuity is fairly constant within the central 2° (visual angle) and drops approximately linearly from there to the 5° foveal border. Beyond 5° acuity drops sharply (approximately exponentially). At scotopic light levels (nighttime) acuity is poor at all eccentricities. We considered an eccentric visual acuity function at photopic light levels reported by Davson³ as a starting point for model resolution degradation when viewing a model at a constant distance.

To model resolution degradation as an object recedes, we also considered visual resolution as a function of stimulus distance (depth). Visual resolution is necessarily dependent upon the quality of its optical components, and variations with stimulus distance result from inaccuracies in accommodation (focusing).¹² The correction of accommodative errors effectively eliminates these variations. Interferometric resolution studies have shown that the effect is dioptric and it is highly likely that the accommodation mechanism is the specific component of visual optics responsible for the phenomenon. We therefore hypothesized that relative spatial acuity diminishes at increased depth. To arrive at an initial estimate of depth scale factor, we considered the National Bureau of Standards (NBS) resolution chart. This chart is constructed of square wave gratings over a wide range of spatial frequencies based on a sequence described by the equation, $\sigma(i) = \sigma(1)r^{i-1}$ in cycles/unit distance, where $\sigma(i)$ is the spatial frequency of the i^{th} pattern, $\sigma(1)$ is the spatial

frequency of the first pattern, and r is the size ratio.²⁶ The spatial frequency of the NBS chart ranges from 1.0 c/mm to 18 c/mm, with $\sigma(1) = 1.0$ c/mm and $r = 1.122$. The chart is constructed by reducing the size of successive square gratings by factor r^{i-1} . We therefore expected visual acuity, and hence model resolution, to degrade with distance at a similar rate. Figure 9 shows the expected acuity function.

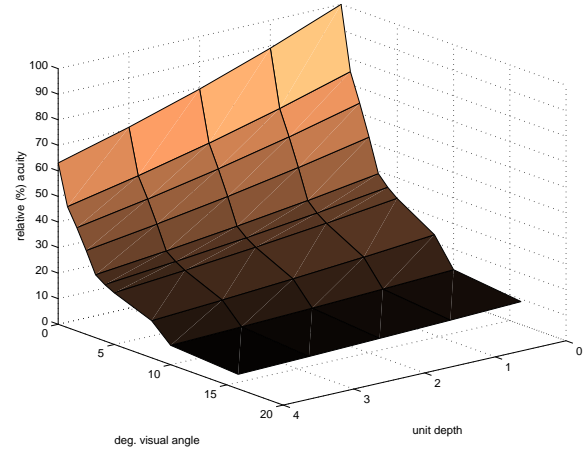


Figure 9: Expected 3D acuity function. The diagram is plotted with unit depth (bot. right axis) vs. deg. visual angle (bot. left axis), where cell shading indicates % rel. acuity as a fn. of eccentricity and distance, with brightest value indicating 100% at $(0, 0)$, the top-most cell.

6.1. Participants

Four participants with a broad range of experience were recruited in an attempt to reduce bias induced by foreknowledge of VR and eye tracking. The group consisted of 2 males and 2 females between the ages of 21 and 47, all with normal or corrected to normal vision. Participants were individually briefed on equipment use prior to the experiment and a non-technical explanation of the graphical system was given to familiarize them with the effects they were about to observe and evaluate.

6.2. Procedure

Each participant was immersed in a virtual environment containing a single multiresolution object. A short 5-point eye tracker calibration sequence was performed at the beginning of each trial. Following a modified two-alternative-forced-choice (2AFC) (method of limits) threshold detection paradigm, the participant was asked to focus on the object and to respond verbally to changes in the appearance of the object when prompted. Between responses, the experimenter manipulated the degradation function by gradually increasing the resolution level until a detection threshold was reached (the participant reported not perceiving further increases in resolution). To reduce the possibility of error

in threshold estimation, a staircase method of stimulus presentation was used, in which the experimenter would first present a series of the stimulus in increasing resolution until the observer changed their response. The series would then be reversed, tracking the stimulus back and forth across the observer’s threshold. This process was repeated at 5 successive levels of depth. Assuming the field of view of the HMD is $75.3^\circ \times 58.4^\circ$ visual angle,³⁰ the object was successively displayed to subtend 26° , 24° , 22° , 20° , and 18° visual angle.

6.3. Discussion

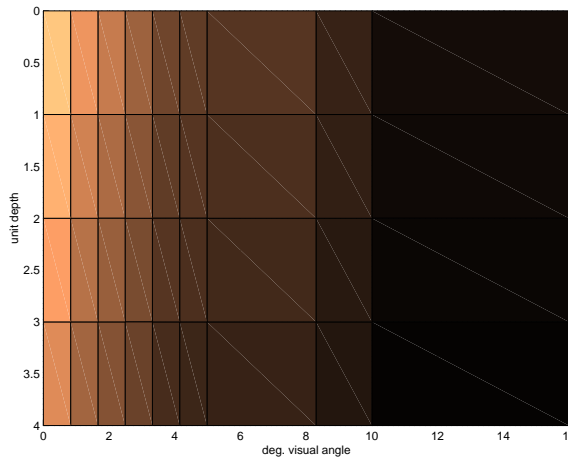
The extents of resolution levels of the empirically derived degradation function are shown in Table 6.3. A comparison of the empirically determined degradation function with expected theoretical acuity limits is shown in Figure 10. It should be noted that the theoretical acuity function relies on a uniform relative acuity scale, whereas the empirical function is nonuniformly distributed on the same scale. Figure 10(b) was constructed by resampling the data in Table 6.3 to match the scale of the theoretical visual acuity function in Figure 10(a). The two functions plots are simi-

Depth	Level 5 (fine)	Level 4	Level 3	Level 2	Level 1 (coarse)
0	2.5°	6.0°	9.8°	14.5°	15.5°
1	1.0°	6.3°	8.5°	14.0°	14.5°
2	0.0°	4.3°	9.5°	11.8°	13.3°
3	1.0°	5.0°	9.5°	10.8°	11.8°
4	0.0°	3.5°	8.3°	13.5°	11.8°

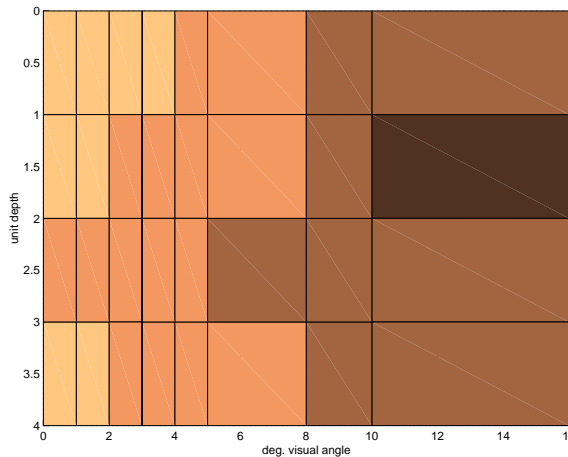
Table 1: *Deg. fcn.: extent of res. levels (deg. vis. angle).*

lar in terms of spatial resolution degradation (shown as cell shading), with an obvious trend within depths (rows) towards decreased resolution at increased visual angle. This trend is also noticeable between depths (columns), suggesting that as the object recedes, and hence diminishes in size, less detail is required. One anomaly in the data is the Level 2 geometry at Depth 4: this is most likely due to the small sample size of the experiment.

Two qualitative observations relevant to future gaze-contingent applications were noted during the course of the experiment, both stemming from subjects’ reports of being aware of peripheral motion. At least two characteristics of the HVS are cooperatively responsible for this phenomenon: selective visual attention, which allows one to attend to objects outside the foveal ROI, and the acute motion sensitivity of the peripheral visual system. To our knowledge, no adequate methods currently exist for tracking attentional deviations from the POR; this would require tracking of both eye motion and brain functions. Recent innovations combining MRI brain imaging and eye tracking technologies may



(a) Theoretical (relative % acuity).



(b) Empirical (relative % resolution).

Figure 10: *Comparison of empirical resolution and theoretical acuity degradation functions. The diagrams are plotted with unit depth (ordinate) vs. deg. visual angle (abscissa), where cell shading indicates % rel. acuity or resolution, with brightest value indicating 100% at (0, 0), the top-left cell.*

eventually become suitable for this purpose. To provide imperceptible peripheral degradation in gaze-contingent systems, the motion sensitivity of peripheral vision must be addressed. Our spatial degradation method alone appears to be insufficient for this purpose; a complete gaze-contingent rendering solution may benefit from a hybrid screen- and model-based approach.

7. Results

After empirically determining the resolution degradation function, system performance was evaluated in terms of polygon counts and frame rates during gaze-contingent viewing. Minima, maxima, and averages are shown in Table 7. The data shows frame rate increases of at least a factor of 2, with up to a 15-fold gain in performance when dis-

object	# of faces	frame rate	gaze-contingent polygon count			gaze-contingent frame rate		
			min. # faces	max. # faces	avg. # faces	min. fr. rate	max. fr. rate	avg. fr. rate
<i>torus</i>	16384	82	96	2272	899	19	4310	1628
<i>igea</i>	268686	4	168	42300	16930	8	2386	309

NOTES

Average face counts and average frame rates are over typical viewing sequence. Maximum frame rate values should be clamped to appropriate display frequency.

Table 2: System performance.

playing the *igea* model (when the maximum frame rate is clamped to the HMD refresh rate of 60Hz). Combining the results with measurements made for the *torus*, the overall average frame rate improvement is approximately 4-fold in favor of gaze-contingent viewing.

Perhaps the most striking performance gains were observed during a comparison of viewing a scene with 24 *igea* meshes at different locations. Full resolution rendering of the scene resulted in frame rates too low to measure, while gaze-contingent viewing allowed near-interactive frame rates (about 20-30 fps).

8. Conclusions

We have described an operational platform for real-time gaze-contingent nonisotropic rendering of multiresolutional geometric objects. The platform is based on a high-end graphics engine and an electromagnetically tracked helmet equipped with a binocular eye tracker. Tracking software obtains helmet position and orientation in real-time and calculates the direction of the user's gaze. The geometric modeling technique developed for the purpose of gaze-contingent rendering includes an integrated approach to tiling, mapping, and remeshing of closed surfaces. A three-dimensional spatial degradation function, determined empirically from human subject experiments, was found to match the expected theoretical limits of the Human Visual System in terms of spatial acuity at varying eccentricity and depth. The empirical spatial degradation function was subsequently used to imperceptibly display multiresolution objects in a gaze-contingent manner. System performance measurements indicate an overall 4-fold average frame rate improvement during gaze-contingent viewing.

Acknowledgments

This work was supported in part by a University Innovation grant (# 1-20-1906-51-4087) and NSF CAREER award # 9984278.

References

1. CLARKE, J. H. Hierarchical Geometric Models for Visible Surface Algorithms. *Communications of the ACM* 19, 10 (October 1976), 547–554. 3

2. DANFORTH, R., DUCHOWSKI, A., GEIST, R., AND MCALILEY, E. A Platform for Gaze-Contingent Virtual Environments. In *Smart Graphics (Papers from the 2000 AAAI Spring Symposium, Technical Report SS-00-04)* (Menlo Park, CA, 2000), AAAI, pp. 66–70. 3
3. DAVSON, H. *Physiology of the Eye*, 4th ed. Academic Press, New York, NY, 1980. 7
4. DUCHOWSKI, A., SHIVASHANKARAIAH, V., RAWLS, T., GRAMOPADHYE, A., MELLO, B., AND KANKI, B. Binocular Eye Tracking in Virtual Reality for Inspection Training. In *Eye Tracking Research & Applications Symposium* (Palm Beach Gardens, FL, 2000), ACM, pp. 89–96. 2, 5
5. DUCHOWSKI, A. T. Acuity-Matching Resolution Degradation Through Wavelet Coefficient Scaling. *IEEE Transactions on Image Processing* 9, 8 (August 2000), 1437–1440. 2
6. DUCHOWSKI, A. T., AND VERTEGAAL, R. *Course 05: Eye-Based Interaction in Graphical Systems: Theory & Practice*. ACM SIGGRAPH, New York, NY, July 2000. SIGGRAPH 2000 Course Notes, URL: <<http://www.vr.clemson.edu/eyetracking/sigcourse/>>, last accessed 9/7/00. 2, 5
7. ECK, M., DEROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERRY, M., AND STUETZLE, W. Multiresolution Analysis of Arbitrary Meshes. In *Computer Graphics (SIGGRAPH '95)* (New York, NY, 1995), ACM, pp. 173–182. 3, 5
8. HELD, R., AND DURLACH, N. Telepresence, time delay and adaptation. In *Pictorial Communication in Virtual and Real Environments*, S. R. Ellis, M. Kaiser, and A. J. Grunwald, Eds. Taylor & Francis, Ltd., London, 1993, pp. 232–246. 2
9. HOPPE, H. View-Dependent Refinement of Progressive Meshes. In *Computer Graphics (SIGGRAPH '97)* (New York, NY, 1997), ACM. 3
10. HORN, B. K. P. *Robot Vision*. The MIT Press, Cambridge, MA, 1986. 5
11. JACOB, R. J. What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Hu-*

- man Factors in Computing Systems: CHI '90 Conference Proceedings (1990), ACM Press, pp. 11–18. 2
12. JOHNSON, C. A. Effects of Luminance and Stimulus Distance on Accommodation and Visual Resolution. *Journal of the Optical Society of America* 66, 2 (February 1976), 138–142. 7
 13. KOCIAN, D. Visual World Subsystem. In *Super Cockpit Industry Days: Super Cockpit/Virtual Crew Systems* (Air Force Museum, Wright-Patterson AFB, OH, 31 March–1 April 1987), Air Force Systems Command/Human Systems Division/Armstrong Aerospace Medical Research Laboratory. 2
 14. KORTUM, P., AND GEISLER, W. S. Implementation of a foveated image coding system for bandwidth reduction of video images. In *Human Vision and Electronic Imaging* (Bellingham, WA, January 1996), SPIE, pp. 350–360. 2
 15. LEVOY, M., AND WHITAKER, R. Gaze-Directed Volume Rendering. In *Computer Graphics (SIGGRAPH '90)* (New York, NY, 1990), ACM, pp. 217–223. 3, 3
 16. LINDSTROM, P., KOLLER, D., RIBARSKY, W., HODGES, L. F., FAUST, N., AND TURNER, G. A. Real-Time, Continuous Level of Detail Rendering of Height Fields. In *Computer Graphics (SIGGRAPH '96)* (New York, NY, 1996), ACM, pp. 109–118. 3
 17. LONGRIDGE, T., THOMAS, M., FERNIE, A., WILLIAMS, T., AND WETZEL, P. Design of an Eye Slaved Area of Interest System for the Simulator Complexity Testbed. In *Interservice/Industry Training Systems Conference* (Brooks Air Force Base, TX, 1989), T. Longridge, Ed., Air Force Human Resources Laboratory, Air Force Systems Command, pp. 275–283. 2
 18. LOSCHKY, L. C., AND MCCONKIE, G. W. User Performance With Gaze Contingent Multiresolutional Displays. In *Eye Tracking Research & Applications Symposium* (Palm Beach Gardens, FL, 2000), ACM, pp. 97–103. 2
 19. LUEBKE, D., HALLEN, B., NEWFIELD, D., AND WATSON, B. Perceptually Driven Simplification Using Gaze-Directed Rendering. Tech. Rep. CS-2000-04, University of Virginia, 2000. 3
 20. LUEBKE, D., VARSHNEY, A., COHEN, J., WATSON, B., AND REDDY, M. *Course 41: Advanced Issues In Level Of Detail*. ACM SIGGRAPH, New York, NY, 2000. SIGGRAPH 2000 Course Notes. 1
 21. MACCRACKEN, R., AND JOY, K. Free-From Deformations With Lattices of Arbitrary Topology. In *Computer Graphics (SIGGRAPH '96)* (New York, NY, 1996), ACM, pp. 181–188. 3
 22. OHSHIMA, T., YAMAMOTO, H., AND TAMURA, H. Gaze-Directed Adaptive Rendering for Interacting with Virtual Space. In *Proceedings of VRAIS'96* (March 30–April 3 1996), IEEE, pp. 103–110. 3
 23. PARKHURST, D., CULURCIELLO, E., AND NIEBUR, E. Evaluating Variable Resolution Displays with Visual Search: Task Performance and Eye Movements. In *Eye Tracking Research & Applications Symposium* (Palm Beach Gardens, FL, 2000), ACM, pp. 105–109. 2
 24. REDDY, M. Specification and Evaluation of Level of Detail Selection Criteria. *Virtual Reality: Research, Development and Application* 3, 2 (1998), 132–143. 2
 25. SCHMALSTIEG, D., AND SCHAUFLEER, G. Smooth Levels of Detail. In *Proceedings of VRAIS'97* (March 1–5 1997), IEEE, pp. 12–19. 3
 26. SMITH, G., AND ATCHISON, D. A. *The Eye and Visual Optical Instruments*. Cambridge University Press, Cambridge, UK, 1997. 7
 27. STARKER, I., AND BOLT, R. A. A Gaze-Responsive Self-Disclosing Display. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings* (1990), ACM Press, pp. 3–9. 2
 28. TANRIVERDI, V., AND JACOB, R. J. K. Interacting with Eye Movements in Virtual Environments. In *Human Factors in Computing Systems: CHI 2000 Conference Proceedings* (2000), ACM Press, pp. 265–272. 2
 29. VINCE, J. A. *Virtual Reality Systems*. Addison-Wesley, Reading, MA, 1995. 3
 30. WATSON, B., WALKER, N., AND HODGES, L. F. Managing Level of Detail through Head-Trackable Peripheral Degradation: A Model and Resulting Design Principles. In *Virtual Reality Software & Technology: Proceedings of the VRST'97* (1997), ACM, pp. 59–63. 8
 31. WATSON, B., WALKER, N., HODGES, L. F., AND WORDEN, A. Managing Level of Detail through Peripheral Degradation: Effects on Search Performance with a Head-Mounted Display. *ACM Transactions on Computer-Human Interaction* 4, 4 (December 1997), 323–346. 2
 32. ZORIN, D., AND SCHRÖDER, P. *Course 23: Subdivision for Modeling and Animation*. ACM SIGGRAPH, New York, NY, 2000. URL: <http://www.mrl.nyu.edu/dzorin/sig00course/> (last accessed 12/30/00). 1, 3
 33. ZORIN, D., SCHRÖDER, P., AND SWELDENS, W. Interactive Multiresolution Mesh Editing. In *Computer Graphics (SIGGRAPH '97)* (New York, NY, 1997), ACM. 3