Gaze-Contingent Displays: Review and Current Trends

Andrew T. Duchowski* Computer Science
Nathan Cournia* Computer Science
Hunter Murphy* Computer Science
Clemson University

(a) Normal vision. (b) As viewed by one with AMD. (c) Pixel-shaded approximation. (d) Visual field mask.

Figure 1: Arbitrary visual field simulation of Age-related Macular Degeneration (AMD): (a) and (b) are from National Institutes of Health [2003], (c) shows our real-time rendering approximation with the visual field mask used in (d).

Abstract

Gaze-Contingent Displays (GCDs) attempt to balance the amount of information displayed against the visual information processing capacity of the observer through real-time eye movement sensing. Based on the assumed knowledge of the instantaneous location of the observer's focus of attention, GCD content can be "tuned" through several display processing means. Screen-based displays alter pixel-level information generally matching the resolvability of the human retina in an effort to maximize bandwidth. Model-based displays alter geometric-level primitives along similar goals. Attentive User Interfaces (AUIs) manage object-level entities (e.g., windows, applications) depending on the assumed attentive state of the observer. Such real-time display manipulation is generally achieved through non-contact, unobtrusive tracking of the observer's eye movements. This paper briefly reviews past and present display techniques as well as emerging graphics and eye tracking technology for Gaze-Contingent Display development.

*{andrewd | acnatha | hmurphy}@vr.clemson.edu

1 Background

Gaze-Contingent Displays (GCDs) degrade the resolution of peripheral image regions in order to reduce computational effort during image transmission, retrieval, or display. In gaze-contingent implementations, the high resolution region moves with the user's focus of attention. An eye tracker is typically used to track the user's gaze. GCDs help increase display speed through compression of peripheral image information, which is not resolvable by the user. Applications include flight and driving simulators, virtual reality, infrared and indirect vision, remote piloting, robotics and automation, teleoperation, and telemedicine; image transmission and retrieval, and video teleconferencing [Baudisch et al. 2003]. In addition to these applications, gaze-contingent displays extend the "moving window" experimental paradigm [McConkie and Rayner 1975] and have thus been invaluable for the purpose of studying visual perception. By removing information beyond perceptual limits, GCDs match the resolvability of human vision, providing compelling visualizations of visual field defects. Thus GCDs can be used to educate students, physicians and patients' family members about the perceptual and performance consequences of vision loss [Geisler and Perry 2002]. Figure 1(b) shows a visualization of Age-related Macular Degeneration (AMD) (vs. normal vision shown in Figure 1(a)) used in a pamphlet issued by the National Institutes of Health (NIH). To render the image, National Eye Institute (NEI) doctors asked their patients with visual impairments what they see and try to get an in-depth description from them. Simulations are then created by computer staff and the doctors have them make changes until they feel that

the information is correct [National Eye Institute 2004]. Although the rendering appears somewhat implausible, as the degenerative area appears to be inverted, image-based GCD techniques described herein could easily generate such a depiction given an appropriate degradation function and fragment program. A simple but plausible resolution degradation function, shown in Figure 1(d), was used to visualize AMD in Figure 1(c).

Duchowski [2003] refers to gaze-contingent display processing as either *screen-based* or *model-based* where the former depends on image processing and the latter on processing graphics primitives (e.g., triangles). GCD research has progressed from simple image-based stimuli (e.g., sine-wave gratings in perceptual research) to complex image-based stimuli (images and video), and more recently to model-based stimuli (e.g., 3D graphical models). Generalizing on this concept, Attentive User Interfaces, or AUIs, control arbitrary objects concommitantly with the user's tracked attentional focal point. Objects may be virtual, such as user interface components on a typical desktop interface (e.g., windows, mouse cursor, etc.) or they may be physical, such as desktop lamps, or television sets.

GCD development has thus progressed from the simple to the complex. In this paper, GCDs are reviewed in reverse order since once again, technological advancement is revolutionizing GCD design at the simplest levels (i.e., image-based GCDs). Thus, the paper reviews progress in Attentive User Interface design, model-based graphical displays, and image-based displays. Image-based are subdivided into focus plus context screens and screen-based GCDs. The paper then introduces recently developed hardware-accelerated techniques for image-based displays and concludes by presenting commodity-off-the-shelf state-of-the-art eye tracking technology.

2 Attentive User Interfaces

Attentive User Interfaces, or AUIs, are an instance of *Non-Command Interfaces* [Jacob 1993] where screen objects, or physical devices, are controlled by gaze. These interfaces rely on methods of user input other than the keyboard or mouse. An example of an eye-slaved interface, often providing a means of communication for quadriplegics, uses the eyes for cursor positioning (e.g., see Majaranta and Raiha [2002] for a comprehensive review of eye typing). By monitoring users' physical proximity, body orientation, and eye fixations, AUIs can be used to control physical objects such as light fixtures and television sets [Shell et al. 2003]. Figure 2 shows several such devices, each equipped with an eyeCONTACT sensor developed by Shell et al. The eyeCONTACT sensor is inexpensive, unobtrusive, tolerant to user head movement, and requires no calibration. It merely detects whether the user is looking toward the sensor. Given this capability, a device equipped with such a sensor can be made "aware" of the user's attentive state. Thus, in the example of the attentive television, if the user is not gazing at the screen, program playback is suspended until viewing is resumed.

3 Model-Based Graphical Displays

In graphical systems, model-based methods aim at reducing resolution by directly manipulating graphical model geometry prior to rendering. Real-time, or gaze-contingent, model manipulation is gaining importance particularly for the benefit of display speedup in immersive displays (e.g., Virtual Reality, or VR) or complex graphical environments (e.g., composed of voluminous data such as millions of triangles). In immersive displays, simplification of the resolution of geometric objects as they recede from the viewer (e.g., in a sense, the distant periphery), as originally proposed by Clarke [1976], is now standard practice, particularly in real-time VR applications. Clarke's original criteria of using the projected area covered by the object for descending the object's Level Of Detail (LOD) hierarchy is still widely used today. However, 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*, object resolution degradation.

A key question regarding LOD control of graphical objects is whether geometry degradation is worth the trouble. That is, this question addresses the tradeoff between resolution degradation and hence rendering time versus any noticeable impact for the user, be it perceptual or performance-based. Recently, Parkhurst and Niebur [2004] evaluated two perceptually adaptive rendering techniques, one velocity-dependent and one gaze-contingent. Decreasing gaze-contingent peripheral geometric detail was found to increase object detection reaction times. Reaction times to localize a target, however, decreased. This suggests that isotropic gaze-contingent LOD impedes target identification while the resultant increased frame rate facilitates virtual interaction.

Isotropic object degradation is not always desirable, however, 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 central $2-5^\circ$, object resolution need not be uniform. Due to the advancements of multiresolution modeling techniques, and to the increased affordability of eye trackers, it is feasible to extend the LOD approach to gaze-contingent displays, where models are rendered *nonisotropically*.

For environments containing significant topological detail, such as virtual terrains or complex objects, rendering with multiple levels of detail, where the level is based on user position and gaze direction, is essential to provide an acceptable combination of surface detail and frame rate.

Figure 2: Attentive User Interfaces: attentive TV with eyeCONTACT sensor (upper inset), light fixture with eyeCONTACT sensor (middle inset), eyePROXY (lower inset). Image courtesy of Roel Vertegaal, from Shell et al. [2003] © 2003 ACM, Inc.

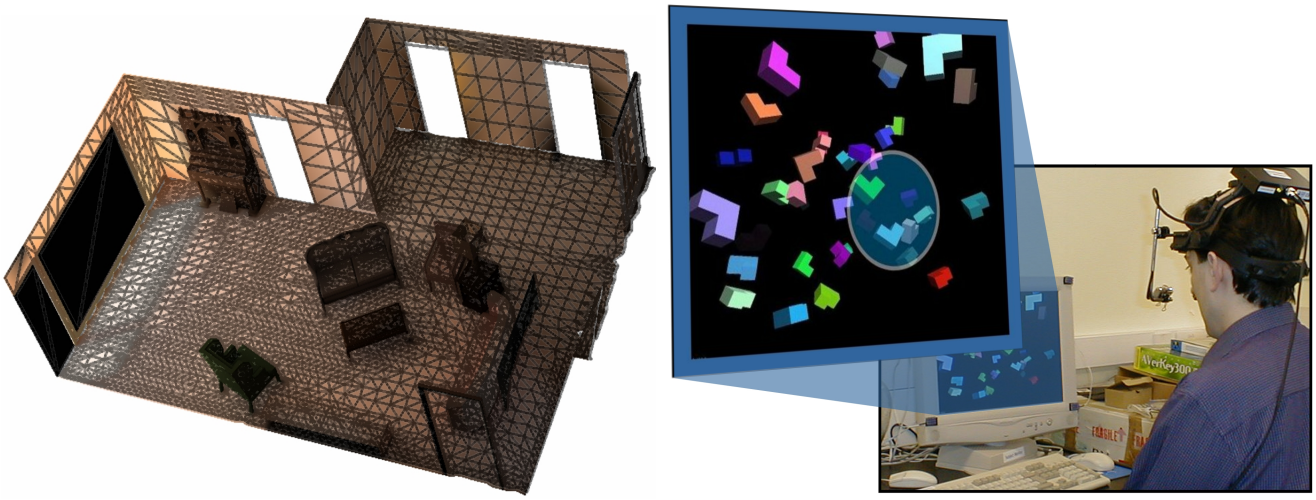


Figure 3: Gaze-contingent spatial and temporal LOD modeling. As the viewer focuses outside the room at the left of the rendering (image at left, courtesy of David Luebke), scene objects located at the right side of the room are rendered using a lower level of spatial detail, indicated by larger triangles (overlayed). Collisions between L-shaped objects (image at right, courtesy of Carol O'Sullivan and John Dingliana) are calculated at a higher level of temporal detail if located within the user's current focus of attention.

One prominent example of an attentive 3D rendering engine varies the LOD at which an object is drawn based on the user's gaze [Luebke et al. 2002]. This way, unattended scene objects are modeled with fewer polygons, even when they are not distant in the scene. Employing a table-mounted monocular eye tracker to measure the viewer's real-time location of gaze over a desktop display, gaze-contingent LOD reduction was found to lead to substantial performance improvements. In the example shown in Figure 3 (left), a reduction of the number of triangles by 70% still leads to an imperceptibly degraded display [Luebke et al. 2002].

Similar LOD degradation benefits have been measured when the graphical scene is displayed within a Head-Mounted Display [Murphy and Duchowski 2001]. A three-dimensional spatial degradation function was obtained from human subject experiments in an attempt to imperceptibly display spatially degraded geometric objects. System performance measurements indicate an approximate overall 10-fold average frame rate improvement during gaze-contingent viewing. An example of a model during gaze-contingent viewing is shown in Figure 4. An interesting finding from this type of research is that an object's silhouette edges are particularly significant for perception, while homogeneous (e.g., flat or smooth) interior object regions are not as interesting.

(a) Original image. (b) Anisotropic degradation (centered).

Figure 4: Gaze-contingent viewing of *igea* model. Courtesy of Hunter Murphy.

Another novel approach to gaze-contingent modeling for real-time graphics rendering was taken by O'Sullivan et al. [2002], who considered temporal resolution in the periphery. More precisely, O'Sullivan et al. developed a degradable collision handling mechanism to limit object collision resolution outside the central display region. Highly prioritized object collisions in the central region are allocated more processing time so that the contact model and resulting visual response is more believable. Having previously noted a significant fall-off in collision resolution detection accuracy at about 4° visual angle, O'Sullivan et al. developed a gaze-contingent collision handling system and reported an overall improvement in the perception of the tracked simulation when the central region was synchronized to the viewer's gaze. An example of the system is shown in Figure 3 (right). The circle in the callout indicates the field of 4° visual angle inside which collisions are processed at greater precision. Saving processing time for collisions outside this area allows spending extra processing time on collisions in the user's focus of attention, which results in an overall improvement in the perception of the simulation.

O'Sullivan et al.'s work is important for exploring the manipulation of peripherally degraded temporal resolution. Consideration of resolution degradation for attentive display generation is a complex issue. There are still many directions this research can take, if simply to explore the manner in which peripheral information is degraded. Should one explore spatial, temporal, color, luminance, or contrast resolution degradation? There is no single answer—research is needed along all of these dimensions.

4 Focus Plus Context Screens

A related display variant to GCDs which are not necessarily gaze-contingent but share the foveal/peripheral demarcation are focus plus context screens. Focus plus context screens achieve the high-detail/low-detail effect by combining a large, wall-sized low-resolution display with an embedded high-resolution screen [Baudisch et al. 2002]. The installation shown in Figure 5 uses an LCD inset combined with projection for generating the low-resolution context. The shown version uses a fixed-position high-resolution focus screen; the iconic illustration at the bottom right shows where it is located. The callout shows the difference in resolutions between the focus and the context area. While the focus area offers enough resolution to allow users to see individual cars, the coarse pixels in the context area merely allow seeing larger objects, such as buildings.

In the example shown, the user is inspecting a specific neighborhood on a satellite image of San Francisco. If the user was using a regular-sized monitor showing the same level of detail as the shown setup, only the neighborhood of interest would be visible, without visual context. With residential areas looking very much alike, it would be hard for the user to tell where the shown portion of the satellite image is located within the city, potentially disorienting the viewer. Adding the low-resolution context screen space brings the Bay bridge and the piers into view, providing additional landmarks that simplify orientation. When the user moves the mouse, the entire display content pans, which allows scrolling display content into the focus region in order to make it high resolution.

For tasks involving large maps or detailed chip designs, focus plus context screens were shown to allow users to work from 20 to 35% faster than when using displays with the

same number of pixels, but in homogeneous resolution or with multiple views. For an interactive driving simulation, users' error rates were only a third of those in a competing multiple-view setup [Baudisch et al. 2002].

In applications that continuously draw the user's attention to the focus area, as is the case for example in the driving simulation used in the experiment, focus plus context screens with a fixed position focus succeed, because the display's focus and context regions cover the user's foveal and peripheral vision the same way a corresponding high-resolution screen does. This makes this type of focus plus context screen, which can be built from comparably inexpensive off-the-shelf components, a cost-effective alternative to complex multi-projector high-resolution screens.

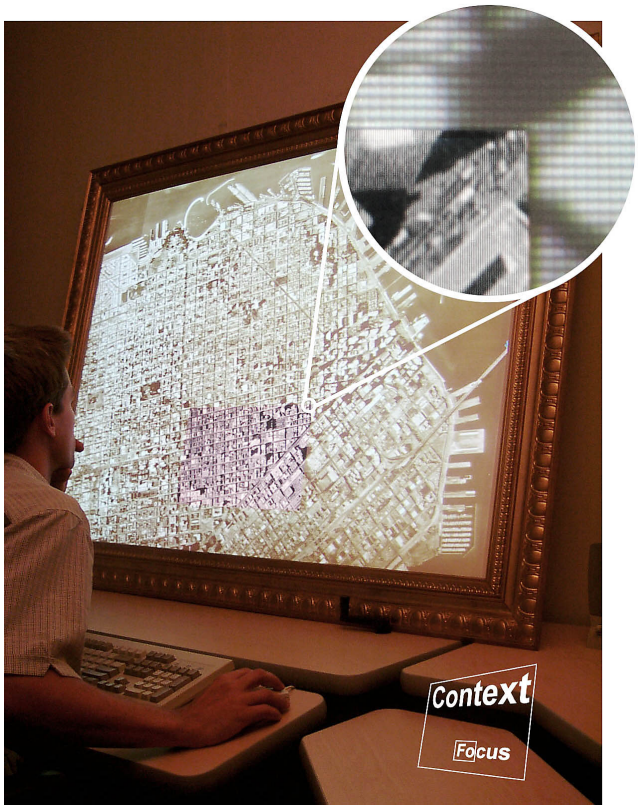


Figure 5: Focus plus context screens consist of a large low-resolution display with an embedded hi-resolution screen. The iconic illustration (bottom right) shows the location of the high-resolution focus screen. The callout shows the difference in resolutions between the focus and the context area. From Baudisch et al. [2003] © 2003 ACM, Inc.

Focus plus context screens are effectively large bi-resolution displays. Idelix, a company that specializes in developing a novel variant of a type of focus plus context screen, has produced Pliable Display Technology, or PDT. The PDT differs from bi-resolution focus plus context screens since instead of providing the traditional foveo-peripheral resolution demarcation, the PDT preserves the periphery at the image's original detail while magnifying the foveal region. An example of a PDT image is shown in Figure 6. Magnification necessarily reduces the spatial resolution of the image beneath the foveal "lens", however, in return the foveal portion of the image shows additional contextual detail due to its magnification.

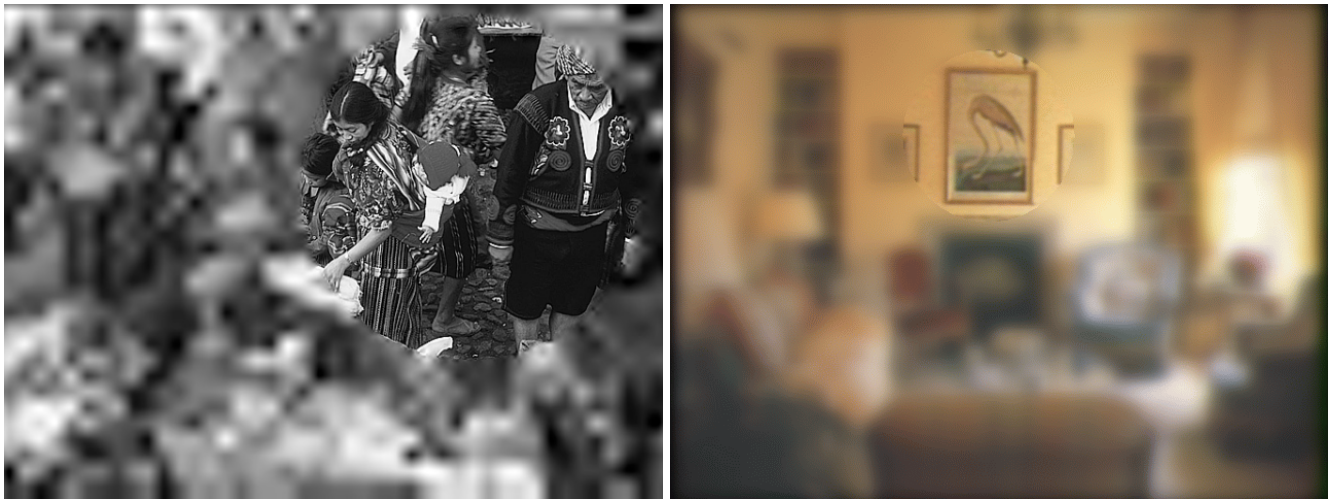
In a sense, the PDT is the reverse of a GCD in terms of resolution. With the PDT motion slaved to a viewer's eye movements, a gaze-contingent PDT offers a novel approach to GCD design. As such, although its benefits to tasks such as visual search (e.g., "find the aircraft in the image" as suggested in Figure 6) are intuitively tantalizing, formal evaluation of this technology is required. Indeed there are numerous questions concerning GCDs in general that can be studied. Examples include the shape of foveal window, shape of the peripheral degradation function, as well as the best technical approach to the display problem, e.g., pixel- or graphics-based (see below). Another salient question is one of usability—in which tasks can GCDs help the viewer?

5 Screen-Based Displays

Extending the idea of foveo-peripheral resolution management exhibited by focus plus context screens, resolution management can be made dynamic if (1) the user's gaze can be measured (e.g., by an eye tracker), and (2) the central high resolution region can be made to move with the user's focus of attention. Gaze-contingent displays have been studied for some time for the purposes of perceptual research (e.g., measurement of the user's perceptual span) and for measurements of system optimization due to compression of peripheral information. Today's improvements in eye tracking and imaging and graphics hardware fuel gaze-contingent display research by allowing researchers to vary information along multiple dimensions, e.g., spatial, temporal, and color resolution.

An experiment conducted by Loschky and McConkie [2000] on a gaze-contingent display investigated spatial, resolutional, and temporal parameters affecting perception and performance. Two key issues addressed by Loschky and McConkie are the timing of GCDs and the detectability of the peripherally degraded component of the GCD. That is, how soon after the end of an eye movement does the window need to be updated in order to avoid disrupting processing, and is there a difference between the window sizes and peripheral degradation levels that are visually detectable and those that produce behavioral effects? In all experiments, monochromatic photographic scenes were used as stimuli with a circular, high-resolution window surrounded by a degraded peripheral region (see Figure 7(a)). Considering temporal update, Loschky and McConkie found that for an image change to go undetected, it must be started within 5 ms after the end of an eye movement. Detection likelihood rose quickly beyond that point. Concerning detection

Figure 6: Application of eye-slaved PDT lens: original *runway* image (left), with magnified region (right).



(a) From Loschky and McConkie [2000] © 2000 ACM, Inc. (b) From Parkhurst et al. [2000] © 2000 ACM, Inc.

Figure 7: Example gaze-contingent screen-based displays.

of peripheral degradation, results showed that the least peripheral degradation went undetected even at the smallest window size (2°), where the opposite was true with the highest level of degradation—it was quite detectable at even the largest window size (5°). The GCD was also evaluated in terms of performance effects, in the context of visual search and scene recall tasks. It was found that the generation of an imperceptible GCD was quite difficult in comparison to the generation of a GCD which does not deteriorate performance. While greater delays (e.g., 15 ms) and greater degradation produce detectable visual artifacts, they appear to have minimal impact on performance of visual tasks when there is a 4.1° high-resolution area centered at the point of gaze. Loschky and McConkie's study shows the importance of considering the intended task for which the display will be used: is the task concerned with perceptual fidelity or visual performance? This is a crucial distinction since although peripheral degradation may be quite noticeable (and hence detrimental to perception), it may not interfere with performance (and thus a benefit to system resource allocation).

Measuring reaction time and accuracy (among other metrics) during a visual search task, Parkhurst et al. [2000] investigated behavioral effects of a two-region gaze-contingent display. Parkhurst et al.'s primary finding is that reaction time and accuracy co-vary as a function of the central region size. The authors note this as a clear indicator of a strategic speed/accuracy tradeoff where participants favor speed in some conditions and accuracy in others. For small central region sizes, slow reaction times are accompanied by high accuracy. Conversely, for large central regions sizes, fast reaction times are accompanied by low accuracy. A secondary finding indicated that fixation duration varies as a function of central region size. For small central region sizes, participants tend to spend more time examining each fixation than under normal viewing conditions. For large central regions, fixation durations tend to be closer to normal. In agreement with reaction time and accuracy, fixation duration is approximately normal (comparable to that seen for uniform resolution displays) with a central region size of 5°. This suggests that the size of the foveal window matters—with a smaller window, users are slower but more accurate, and vice versa.

For screen-based VR rendering the work of Watson et al. [1997] is particularly relevant. Watson et al. studied the effects of Level Of Detail (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.

Traditional metrics for screen-based GCDs have considered peripheral degradation (typical in terms of spatial or contrast resolution), at threshold. Watson et al.'s [2004] most recent evaluation of peripheral LOD control considers supra-threshold perception. Specifically, Watson et al. report that LOD must support a task-dependent level of perceptibility. Below this level, LOD should *increase* when eccentricity is high or contrast is low, and all scales of LOD (fine or coarse) are equally important.

Recently, GCDs have been developed to incorporate arbitrary resolution maps, supporting foveal regions of arbitrary shape. This has allowed the generation of high quality images with minimal artifacts at real-time display frame rates. Geisler and Perry [1998] describe a multi-resolution pyramidal method for creating variable resolution displays in real-time using general-purpose computers. Foveal regions (more than one can be defined) can be created to vary in shape and size. The system generates high quality images (minimal artifacts) at high (real-time) display frame rates.

Geisler and Perry [2002] extended their method to allow completely arbitrary variable resolution displays. The new version of their software produces artifact-free gaze-contingent video at high frame rates in either 8-bit gray scale or 24-bit color. Geisler and Perry's display depends on pyramidal pre-processing of the images prior to display. Rendering appears to use a graphics card for image display, but the card itself does not appear to be used for image processing.

Given an arbitrary degradation function, as shown in Figure 8, the gaze-contingent display can be used to examine various facets of perception or performance. We expect the flexibility of such displays will facilitate further investigation of attentional principles along multiple dimensions, such as spatiotemporal resolution, contrast, and color.

6 Current Trends

Prior research of image-based gaze-contingent displays has mostly focused on perceptual or performance effects of the reduction of peripheral spatial frequency (i.e., cycles per degree or bits per pixel). For two excellent surveys on GCDs, see Reingold et al. [2003] and Parkhurst and Niebur [2002]. Due to hardware limitations, a good deal of prior work relied on image pre-processing. For gaze-contingent displays, pre-processed images would be recalled from memory on a "just-in-time" basis, i.e., usually in relation to the location of the user's eye tracked so-called Point Of Regard (POR). Due to recent advancements in computer hardware, gaze-contingent imaging researchers have begun utilizing hardware to perform image processing operations in real-time. In a recent example of hardware-accelerated eye-movement controlled image coding, Bergström [2003] used a DCT-based image codec to achieve real-time image compression and display.

In this section, technical aspects are presented of a novel hardware-accelerated approach to gaze-contingent multi-resolution display design for the real-time simulation of arbitrary visual fields using a commodity graphics card. The approach uses mipmapping for dyadic image degradation and

Figure 8: Gaze-contingent display showing a scene from the movie *The Gladiator*. As the user focuses on the face of the shot's main character, all other display content is rendered at reduced resolution. This type of display can be used for gaze-contingent compression purposes or for the study of human visual perception—in this case the display is used to study glaucoma patients. Original image shown in bottom left inset, the arbitrary visual field used to simulate glaucoma is shown at bottom right. Original image © 2000 DreamWorks SKG and Universal Studios; gaze-contingent rendering and resolution map courtesy of Bill Geisler and Jeff Perry.

an arbitrary mask image for creating the foveal/peripheral demarcation.

Mipmapping relies on texture-mapping (and shader programming), which is a hybrid of model- and image-based approaches. Peripheral degradation of the image still relies on image processing, albeit the image is now considered a texture map. Rendering of the image relies on mapping the image onto a simple graphical object, in most cases a polygon (usually a screen aligned quadrilateral) of the same dimension as the display window.

There are several tradeoffs between the texture-mapping and screen-based approaches, although both are now typically provided by graphics libraries such as OPENGL [Shreiner et al. 2003]. Advantages of the screen-based approach include the following:

- Image resolution is of minor importance. Provided the viewing window is made to be the same size as the given image, the resultant display is generally shown at 1:1 pixel mapping, i.e., the image is drawn to scale.

- Provided a graphics card that supports OPENGL's Imaging Subset in hardware is used, image processing operations can be performed quickly via hardware-accelerated convolution.

- Various blending operations are provided that enable simple image combinations to take place via an image's alpha channel.

There are, however, disadvantages to the screen-based approach:

- Not all graphics cards support (or supported) the Imaging Subset in hardware. For example, the NVidia GeForce4 Ti 4600 card did not, but its more expensive cousin the NVidia Quadro4 (e.g., XGL 900) did. Lacking hardware support for the Imaging Subset, imaging operations such as convolution with the GeForce4 reverted to software implementation. This resulted in noticeable speed degradation.

- The most significant drawback of the screen-based approach for gaze-contingent display is that the required image blending functions (for blending foveal and peripheral image portions) rely on the images' alpha channels. Thus, to provide a gaze-contingent display, the image alpha channels would need to be translated in real-time to match the foveal region, a prohibitively expensive operation.

Texture-mapping, and in particular multitexturing and related fragment programming, solves the blending problem since the alpha channel can be dissociated from either foveal or peripheral image and made into its own image. This is an important point since once so dissociated, the alpha mask can be manipulated independently. The manipulation that is most relevant to gaze-contingent display is translation of the foveal mask. Since mask translation is performed quickly in hardware, the result is real-time movement of the foveal region. There are, however, disadvantages to the texture-mapping approach:

- In general, texture mapping is more complicated than simple image drawing since it relies on the definition of the graphical object that is to be textured. Using a quadrilateral for this purpose is often the most simplest and logical choice. Following geometry definition, textures need to be defined, bound, and loaded into memory. There are numerous options for doing so (this is somewhat of a blessing and a curse).

- Because texture mapping generally relies on a geometric primitive, and that primitive is subject to geometric transformations, the resultant display may or may not preserve the 1:1 pixel mapping between original image and final display. In contrast to the Imaging Subset, one usually needs to define the window size (as before), and also the polygon onto which the image will be texture-mapped. Care must be taken to properly display the polygon without inadvertently changing the polygon's size (which is quite easy to do, e.g., via viewing transformations).

- To display the texture-mapped primitive, texture coordinates are required. Care must be taken not to introduce inadvertent image scaling, shifting, etc., through improper coordinate use.

To summarize the distinction between screen-based and texture-based approaches, texture-mapping offers much greater flexibility in image display at the expense of additional complexity.

6.1 Mipmapping

For fully hardware-accelerated display, as discussed here, GCDs can utilize in-hardware image degradation provided by built-in mipmapping functions. Mipmapping provides a method of prefiltering an image (texture) at multiple levels of detail [Williams 1983]. [1] Mipmaps are dyadically (by powers of two) reduced versions of a high-resolution image. One can either create these images manually *a priori*, or have them created automatically by OPENGL. Automatic creation checks to see if image dimensions are a power of 2. If not, a copy of the image data is scaled up or down to the nearest power of 2.

Several filter options are available for generating coarsely subsampled or linearly interpolated images. Four texture minification options control combinations of inter- and intra-map pixel interpolation. The effect of these commands generates a coarsely or smoothly degraded periphery for dyadic

[1] The term *mip* stands for the Latin *multum in parvo*, meaning "many things in a small place" [Williams 1983].

levels of degradation. Real-time control of the texture environment and texture parameters allows on-the-fly switching of peripheral degradation.

Below, two recent approaches based on mipmapping are first briefly reviewed for completeness and comparison to the subsequent newly introduced fragment programming technique. The former approach is suitable for implementations on 3rd-generation graphics cards while the latter strategy requires 4th-generation cards.

6.2 Multitexturing

Real-time rendering of a bi-resolution gaze-contingent display relies on two images. The first requirement is the source image for generating a high-resolution inset as well as a low-resolution background. The low-resolution background image is generated by dyadically degrading in hardware the source image via OpenGL's mipmapping facilities. Alternately, the source image (or another image altogether) may be pre-processed in some other way and can be substituted for the background image. The second required image is an arbitrary visual mask whose shape forms the foveal window.

Using special effects compositing terminology, the mask image simply constitutes the matte image which serves as the alpha mask for blending of the foreground (high-resolution) and background (low-resolution) images. The matte image is typically a normalized greyscale image where pixel values of 1 represent portions of the high-resolution image that show through while values of 0 are masked and therefore replaced by the corresponding background image pixels. Of course, any greyscale image can be used instead to simulate an arbitrary visual field (e.g., to simulate glaucoma or AMD). Simply inverting a Gaussian 1-center, 0-surround map, for example, would result in the "moving mask" paradigm used in perceptual vision research (see Bertera and Rayner [2000]).

To obtain a composited rendering of a foveal high-resolution window atop a low-resolution background, three textures are created for a quadrilateral. The first texture, assigned to Texture Unit 0, or TU0, is the image mask. The second texture is the given image which is assigned as the foreground image at Texture Unit 1, or TU1. The third texture is the original image used for the foreground, also mipmapped, with the exception of the use of different Level Of Detail (LOD). It is the coarser LOD that generates the degraded background in the gaze-contingent display.

During display, the mask texture at TU0 is translated to the real-time coordinates of the foveal position. The process is shown diagrammatically in Figure 9, with the callout showing the change in resolution between foveal and background regions. For printing considerations, a greyscale image is used as the example stimulus although the texture-mapping methodology applies equally well to 24-bit (or 32-bit) color images.

An alternative approach, but also based on multitexturing, involves using two quadrilaterals instead of three,

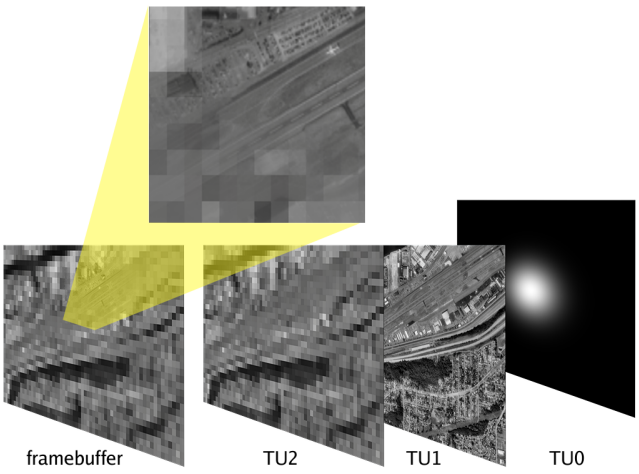


Figure 9: Multitexture blending graphics pipeline.

as shown by Nikolov et al. [2004], who use a similar approach to the above and apply their gaze-contingent display to numerous applications, including gaze-contingent multi-resolution displays, gaze-contingent multi-modality displays (e.g., graphical maps overlayed on aerial photographs), and gaze-contingent image analysis.

6.3 Fragment Programming

The three-texture approach described above leads to a bi-resolutional display. For a more robust (and accurate) representation of human visual acuity, multiple levels of detail are needed in the periphery, resulting in anisotropic peripheral degradation, otherwise known as a Multi-Resolutional Gaze-Contingent Display (MRGCD). To provide multiple levels of resolution in the periphery, the above multitexturing approach would require the use of multiple texture units. What is required is schematically shown in Figure 10 (from Duchowski [1997]). At any given pixel, concentrically related to the foveal position, a lookup is needed to a pixel at a specific level of resolution. Fragment programs provide just this type of flexibility by providing control of mipmap LOD at each fragment (pixel). This is provided by the (undocumented) `tex2Dbias() Cg` call, or `TXB ARB_fragment_program` assembly instruction. The `TXB` instruction takes the first three components of its source vector and maps them to s, t, and r. These coordinates are used to sample from the specified texture target on the specified texture image unit in a manner consistent with its parameters. Additionally, the fourth component of the source vector is applied to equation (1) as $fragment_{bias}$ to further bias the LOD [OpenGL Architectural Review Board 2003]:

$$\lambda(x,y) = log_2[\rho(x,y)] + clamp(texobj_{bias} + texunit_{bias} + fragment_{bias}) \quad (1)$$

The resulting sample is mapped to RGBA and written to the result vector. Unlike multitexturing, this rather elegant ap-

proach does not require explicit blending. Instead, the appropriate mipmap level (bias) is obtained directly at each fragment. Note that if the degradation map is allowed to change dynamically, fragment programming allows dynamic visual field representation, e.g., allowing multiple "Regions Of Interest" (ROIs) which could be used for pre-attentive display purposes [Duchowski and McCormick 1995].

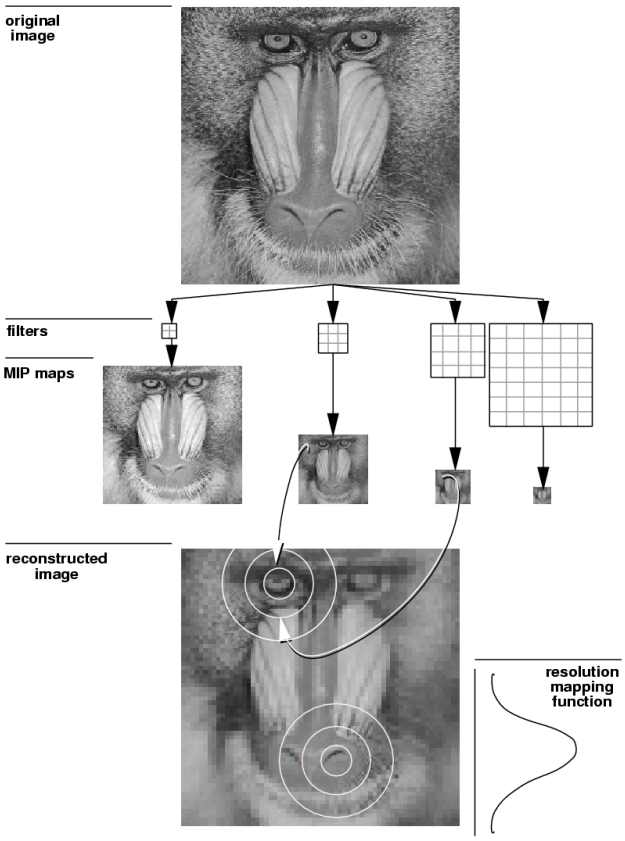


Figure 10: Illustration of per-fragment mipmap LOD bias selection.

Another rather powerful but as yet unexploited benefit of fragment programming is the potential for gaze-contingent color degradation. This is achieved by the use of a 4-channel degradation mask. Since only one channel (the alpha channel) is needed for resolution degradation, it is natural to use the remaining RGB channels to represent color degradation maps. Hence color degradation can be independently controlled in RGB color-space. Since each of the RGB channels is itself a normalized image, color can simply be degraded by scaling the given pixel's color according to the scalar found in the corresponding RGB channels, e.g., (in Cg syntax)

$$\begin{aligned} rgb2grey &= float4(0.299, 0.587, 0.114, 1.0); \\ color &= rgba.xyz * m.xyz + \\ &\quad dot(rgb2grey.xyz, (rgba.xyz * (1 - m.xyz))), \end{aligned} \tag{2}$$

where *rgba* is the texture sample, *rgb2grey* is the constant luminance conversion coefficient vector, and *m* is the arbitrary 4-channel visual field mask. Equation (2) simply interpolates a pixel's output color between its full color (original) and its luminance. Due to the independence of the RGB degradation channels, this offers a rather powerful technique for exploring perceptual effects of peripheral color degradation. While peripheral visual acuity (and contrast sensitivity) have been studied widely, peripheral color acuity has not. Thus the hardware-accelerated fragment programming technique offers considerable flexibility for future perceptual research.

Source code for a simple `GLUT` example is available on the web: <http://andrewd.ces.clemson.edu/gcd/> The current GCD code has been tested via both mouse- and eye-controlled foveal window and runs well above hardware display rates, i.e., 60 fps. The code has also been extended to display video streams by interfacing with a video loading library (`xine-lib`).[2] Because of `gluBuild2DMipmaps()` hardware subsampling of a given image, we have found that the GCD code is sufficiently fast for real-time video degradation (display rates have informally been measured well above 60 fps). This suggests that for gaze-contingent display, image processing no longer poses a significant bottleneck, obviating the need for image pre-processing or storage.

7 Eye Tracking Technology

The above multitexturing and fragment programming techniques for gaze-contingent viewing are presented independent of eye tracker software. To fully implement a GCD, all that is necessary is to equip the main rendering loop with code that obtains the instantaneous x, y coordinates of the the user's gaze and applies these to the required translation of the foveal mask.

Eye tracker technology has advanced significantly since its modern inception in the early 20th century. From the first method of eye tracking using corneal reflection in 1901, through the use of contact lenses in the 1950s, today's eye trackers generally employ analog video-based eye tracking techniques developed circa the 1970s [Duchowski 2003]. Consider eye trackers within the following taxonomy:

1. First generation: eye-in-head measurement of the eye consisting of techniques such as scleral contact lens/search coil, electro-oculography;
2. Second generation: photo- and video-oculography;
3. Third generation: analog video-based combined pupil/corneal reflection.

The most salient form of eye tracking output is estimation of the projected Point Of Regard (POR) of the viewer. First and second generation eye trackers generally did not provide this type of information (the latter almost does but here video-oculography is lumped into second generation systems since, within this taxonomy, eye movement analysis relied

[2] <http://xine.sf.net/>

Table 1: Functional eye tracker comparison.

	legacy systems	state-of-the-art
Technology	analog video	digital video
Calibration	5- or 9-point, tracker-controlled	any number, application-controlled
Optics	requires focusing/thresholding	automatic
Communication	serial	TCP/IP (client/server)
Synchronization	status byte word	API callback

on frame-by-frame visual inspection of photographs or video frames and did not allow easy POR calculation). So-called video-based combined pupil/corneal reflection eye trackers easily provide POR calculation following calibration, and are today *de rigeur*. Due to the availability of fast analog video processors, these third-generation eye trackers are capable of delivering the calculated POR in real-time. However, eye tracking technology is about to undergo its next evolution. Fourth-generation eye trackers, which are just beginning to appear on the market, are starting to make use of digital optics. Coupled with on-chip Digital Signal Processors (DSPs), eye tracking technology stands to significantly increase in usability, accuracy, and speed while decreasing in cost.

While the latter part of the above prediction has not yet materialized, the former three points have. The state of today's technology can best be summarized by a brief functional comparison of equipment available about 5 years ago and today's state-of-the-art given in Table 1.

In general, most eye tracking applications perform the following [Duchowski 2003]:

1. Connection: establish connection with the eye tracker (e.g., serial port or TCP/IP).

2. Calibration: display calibration points at the appropriate location and time.

3. Synchronization: display stimulus at the appropriate time (the eye tracker should be able to inform the application program of its state, or vice versa).

4. Data streaming: use eye tracker to capture data and/or update the stimulus scene in a gaze-contingent manner.

An example of a fourth-generation eye tracker is available from Tobii Technology AB [2003]. The Tobii 1750 eye tracker can be configured in several ways, one of which is acting as a server for a (possibly remote) eye tracking client application. The benefit of this organization is platform independence since communication between client and server occurs over TCP/IP. Platform independence is true for an eye tracker communicating via a serial cable as well, although serial communication requires relatively closer proximity between the eye tracker and application computer. An example configuration with an application computer (e.g., Linux PC) connected to the Tobii eye tracker is shown in Figure 11.

References

BAUDISCH, P., DECARLO, D., DUCHOWSKI, A. T., AND GEISLER, W. S. 2003. Focusing on the Essential: Considering Attention in Display Design. *Communications of the ACM 46*, 3.

BAUDISCH, P., GOOD, N., BELLOTTI, V., AND SCHRAEDLEY, P. 2002. Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming. In *Proceedings of CHI '02*. ACM, Minneapolis, MN, 259–266.

BERGSTRÖM, P. 2003. Eye-movement Controlled Image Coding. Ph.D. thesis, Linköping University, Linköping, Sweden.

BERTERA, J. H. AND RAYNER, K. 2000. Eye Movements and the Span of the Effective Stimulus in Visual Search. *Perception & Psychophysics 62*, 3, 576–585.

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

DUCHOWSKI, A. T. 1997. Gaze-Contingent Visual Communication. Ph.D. thesis, Texas A&M University, College Station, TX.

DUCHOWSKI, A. T. 2003. *Eye Tracking Methodology: Theory & Practice*. Springer-Verlag, Inc., London, UK.

DUCHOWSKI, A. T. AND MCCORMICK, B. H. 1995. Pre-attentive considerations for gaze-contingent image processing. In *Human Vision, Visual Processing, and Digital Display VI (SPIE vol. 2411)*. SPIE, Bellingham, WA, 128–139.

GEISLER, W. S. AND PERRY, J. S. 1998. Real-time foveated multiresolution system for low-bandwidth video communication. In *Human Vision and Electronic Imaging*. SPIE, Bellingham, WA.

GEISLER, W. S. AND PERRY, J. S. 2002. Real-time Simulation of Arbitrary Visual Fields. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, New Orleans, LA, 83–153.

JACOB, R. J. K. 1993. Eye-Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces. In *Advances in Human-Computer Interaction*, H. R. Hartson and D. Hix, Eds. Ablex Publishing Co., Norwood, NJ, 151–190. URL: <http://www.cs.tufts.edu/∼jacob/papers/hartson.ps>, last accessed 9/4/00.

LOSCHKY, L. C. AND MCCONKIE, G. W. 2000. User Performance With Gaze Contingent Multiresolutional Displays. In *Eye Tracking Research & Applications Symposium*. ACM, Palm Beach Gardens, FL, 97–103.

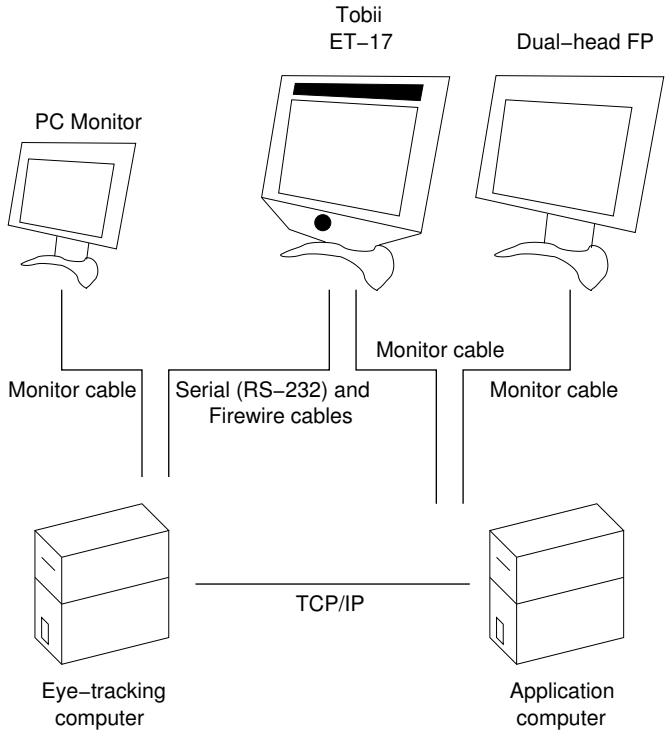
LUEBKE, D., REDDY, M., COHEN, J., VARSHNEY, A., WATSON, B., AND HUEBNER, R. 2002. *Level of Detail for 3D Graphics*. Morgan-Kaufmann Publishers, San Francisco, CA.

MAJARANTA, P. AND RAIHA, K.-J. 2002. Twenty Years of Eye Typing: Systems and Design Issues. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, New Orleans, LA.

MCCONKIE, G. W. AND RAYNER, K. 1975. The Span of the Effective Stimulus During a Fixation in Reading. *Perception & Psychophysics 17*, 578–586.

MURPHY, H. AND DUCHOWSKI, A. T. 2001. Gaze-Contingent Level Of Detail. In *EuroGraphics*. EuroGraphics, Manchester, UK.

NATIONAL EYE INSTITUTE. 2004. Office of Communication, Health Education, Personal communique.



(a) Schematic configuration.

(b) The ET-1750 flat panel.

(c) Typical use.

Figure 11: Single Tobii eye tracking station hardware setup (one possible configuration).

NATIONAL INSTITUTES OF HEALTH. 2003. Age-Related Macular Degeneration: What you should know. Publication No: 03-2294, National Eye Institute, National Institutes of Health, 2020 Vision Place, Bethesda, MD 29892-3655. URL: <http://www.nei.nih.gov/health/maculardegen/armd_facts.htm> (last accessed April 2004).

NIKOLOV, S. G., NEWMAN, T. D., BULL, D. R., CANAGARAJAH, N. C., JONES, M. G., AND GILCHRIST, I. D. 2004. Gaze-Contingent Display Using Texture Mapping and OpenGL: System and Applications. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, San Antonio, TX, 11–18.

O'SULLIVAN, C., DINGLIANA, J., AND HOWLETT, S. 2002. Gaze-Contingent Algorithms for Interactive Graphics. In *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movement Research*, J. Hyöna, R. Radach, and H. Deubel, Eds. Elsevier Science, Oxford, England.

PARKHURST, D., CULURCIELLO, E., AND NIEBUR, E. 2000. Evaluating Variable Resolution Displays with Visual Search: Task Performance and Eye Movements. In *Eye Tracking Research & Applications Symposium*. ACM, Palm Beach Gardens, FL, 105–109.

PARKHURST, D. J. AND NIEBUR, E. 2002. Variable Resolution Displays: A Theoretical, Practical, and Behavioral Evaluation. *Human Factors 44*, 4, 611–629.

PARKHURST, D. J. AND NIEBUR, E. 2004. A Feasibility Test for Perceptually Adaptive Level of Detail Rendering on Desktop Systems. In *Applied Perception and Graphics Visualization (APGV)*. ACM, Los Angeles, CA, to appear.

REINGOLD, E. M., LOSCHKY, L. C., MCCONKIE, G. W., AND STAMPE, D. M. 2003. Gaze-Contingent Multi-Resolutional Displays: An Integrative Review. *Human Factors 45*, 2, 307–328.

OPENGL ARCHITECTURAL REVIEW BOARD. 2003. ARB_FRAGMENT_PROGRAM Specification. Revision: 26. URL: <http://oss.sgi.com/projects/ogl-sample/registry/> (last accessed April 2004).

SHELL, J. S., SELKER, T., AND VERTEGAAL, R. 2003. Interacting with Groups of Computers. *Commun. ACM 46*, 3 (March), 40–46.

SHREINER, D., WOO, M., NEIDER, J., AND DAVIS, T. 2003. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.4*, 4th ed. Addison-Wesley.

TOBII TECHNOLOGY AB. 2003. Tobii ET-17 Eye-tracker Product Description. (Version 1.1).

WATSON, B., WALKER, N., AND HODGES, L. F. 2004. Supra-Threshold Control of Peripheral LOD. *ACM Transactions on Graphics 23*, 3 (July), (to appear).

WATSON, B., WALKER, N., HODGES, L. F., AND WORDEN, A. 1997. 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), 323–346.

WILLIAMS, L. 1983. Pyramidal Parametrics. *Computer Graphics 17*, 3 (July), 1–11.