Chapter 24
Computer Science

This chapter gives an overview of primarily interactive applications mainly developed by computer science researchers. Following the hierarchy of eye tracking systems given earlier, apart from diagnostic usability studies, this chapter focuses on two types of interactive applications: selective and gaze-contingent. The former approach uses an eye tracker as an input device, similar in some ways to a mouse. This type of ocular interaction is often studied by researchers involved in the fields of Human–Computer Interaction (HCI) and Computer-Supported Collaborative Work (CSCW). The latter gaze-contingent application is a type of display system wherein the information presented to the viewer is generally manipulated to match the processing capability of the human visual system, often matching foveo–peripheral perception in real-time. It should be noted that a good deal of previous work discussed in this chapter is based on conference proceedings.

24.1 Human–Computer Interaction and Collaborative Systems

Eye-based interactive systems have been presented at several SIGCHI conferences with a significant increase in the number of papers in recent years. Most of these papers have traditionally focused on interactive uses of eye trackers, although diagnostic applications have also begun to appear, particularly in the context of usability studies.

Interactive uses of eye trackers typically employ gaze as a pointing modality, e.g., using gaze in a similar manner to a mouse pointer. Prominent applications involve selection of interface items (menus, buttons, etc.), as well as selection of objects or areas in Virtual Reality (VR). A prototypical "real-world" application of gaze as an interactive modality is eye typing, particularly for handicapped users.

A.T. Duchowski, *Eye Tracking Methodology*, DOI 10.1007/978-3-319-57883-5_24

Other uses of gaze in the general field of human–computer interaction involve gaze as an indirect pointing modality, for example as a deictic reference in the context of collaborative systems, or as an indirect pointing aid in user interfaces. Diagnostic uses of eye trackers are becoming adopted for usability studies, i.e., testing the effectiveness of interfaces as evidenced by where users look on the display.

24.1.1 Classic Eye-Based Interaction

One of the first eye-based interactive systems, introduced by Jacob, demonstrated an intelligent gaze-based informational display. In Jacob's system a text window would scroll to show information on visually selected items. The paper's title, "What You Look At Is What You Get," is a play on words on the common word processing/printing paradigm of What You See Is What You Get (WYSIWYG). Jacob's paper was an early paper describing the feasibility of a gaze-based interactive system in which the author discussed the possibility of using gaze in place of or in addition to a mouse pointer. The paper was one of the first to use video-based corneal reflection eye tracking technology interactively and is well known for its identification of an important problem in eye-based interactive systems: the Midas Touch problem. Essentially, if the eyes are used in a manner similar to a mouse, a difficulty arises in determining intended activation of foveated features (the eyes do not register button clicks!). That is, unlike a mouse with which a user signifies activation of an object by pressing a mouse button, with gaze pointing everything that a user looks at is potentially activated (and so in the Midas analogy unintentionally turns to gold). To avoid the Midas Touch problem, Jacob discusses several possible solutions including blinks, finally promoting the use of dwell time (of about 150–200 ms) to act as a selection mechanism.

At the same SIGCHI meeting in which Jacob's paper appeared, a graphical "self-disclosing" display was presented by Starker and Bolt (1990). This interactive system provided the user with gaze-controlled navigation in a three-dimensional graphics world. The graphical environment contained story world characters who responded in interesting ways to the user's gaze. Fixations activated object "behaviors", because the system would maintain and increase the user's visual interest level in an object. With increased interest, objects would blush and/or provide a verbal narrative. Unlike Jacob's use of dwell time, in this system dwell time was used to zoom into the graphics world.

At a more recent SIGCHI meeting, Tanriverdi and Jacob (2000) presented a new gaze-based interactive system, this time with gaze acting as a selective mechanism in VR. In this system Tanriverdi and Jacob compared eye-based interaction with hand-based interaction. The authors found that performance with gaze selection was significantly faster than hand pointing, especially in environments where objects were placed far away from the user's location. In contrast, no performance difference was found in "close" environments. Furthermore, altough pointing speed may increase with gaze selection, there appears to be a cognitive tradeoff for this gain in efficiency:

subjects had more difficulty recalling locations they interacted with when using gaze-based selection than when using hand selection.

24.1.2 Cognitive Modeling

In their paper presented at SIGCHI, Byrne et al. (1999) tested the arrangement of items during visual search of click-down menus. The authors contrasted two computational cognitive models designed to predict latency, accuracy, and ease of learning for a wide variety of HCI-related tasks: EPIC (Executive Process Interactive Control; Kieras and Meyer 1995) and Adaptive Control of Thought-Rational (ACT-R; Anderson 2002). The EPIC architecture provides a general framework for simulating a human interacting with her or his environment to accomplish a task (Hornof and Kieras 1997). ACT-R is a framework for understanding human cognition whose basic claim is that cognitive skill is composed of production rules (Anderson 1993). These models specifically predict, in different ways, the relationship between eye and mouse movement. With respect to click-down menus, the EPIC model predicts the following.

1. The distance covered by saccades is constant.
2. Eyes tend to overshoot the target with some regularity.
3. No mouse movement is initiated until the target is visually acquired.
4. Eye movement patterns are generally top-down and random.

In contrast, the ACT-R model predicts:

1. The distance covered by saccades is variable.
2. Eyes never overshoot the target.
3. Mouse movement generally follows saccades.
4. Eye movement patterns are exclusively top-down.

A visual search experiment was conducted where a target item was presented prior to display of a click-down menu. Based on the number of fixations, both models were supported because in some cases visual search exhibited exclusively top-down search (ACT-R) and in others both top-down and random patterns were observed (EPIC). Fixation positions, however, did not appear random, they did not appear to be strictly linear, and there appeared to be a preference for the first item in the menu. This particular study is informative for two reasons. First, it shows the importance of a good model (or lack of one) that can be used for designing user interfaces. If a model can be developed that successfully predicts some aspect of user activity, then this model may be used to design an interface that benefits the user by adapting to the user (instead of, for example, forcing the user to learn a new, possibly unintuitive style of interaction). The idea of modeling human behavior is an important concept in human–computer interaction. Second, the paper points out that eye tracking is an effective usability modality and that a new model for visual search over menus is needed, e.g., possibly a "noisy" version of top-down search.

Beyond keystroke-level modeling and their variants, eye tracking has recently been used in the field of psychology of programming: understanding the cognitive processes involved in debugging and comprehending computer programs. In an effort to develop an eye tracking methodology for studying program comprehension, Bednarik (2005) compared eye tracking to mouse-contingent display, i.e., blurring the display except in the neighborhood of the mouse cursor, as performed in the so-called Restricted Focus Viewer, or RFV technique.

The RFV technique has been suggested in the literature as an alternative to eye tracking, both because it is cheaper (it does not require any hardware in addition to the normal workstation), and because it is unobtrusive and accurate at the same time (in the sense that there is no measurement error (see Tarasewich et al. (2005) for a review of RFV and an ehanced method designed for Web page usability testing). However, it is reasonable to ask how much the blurring of most of the display affects normal viewing behavior and, thereby, possibly disrupting the observations made using the technique. Previous research has implied that the technique could be a valid tool, but the tasks where it has been used have been relatively simple. Bednarik used an eye tracker to repeat an experiment that had previously been performed with RFV and cast doubt on the latter's validity. Addressing the question in detail, controlled experiments showed that an eye tracker clearly outperformed RFV in terms of validity, given the particular task of program debugging. For other tasks, such as Web page usability testing, Tarasewich et al.'s Enhanced RFV technique may be useful, although probably not as a substitute for eye tracking (see Sect. 23.5).

Studying the same cognitive processes (program comprehension), Uwano et al. (2006) provided interesting static visualiztions of dynamic eye movements [similar to those of Räihä et al. (2005)] during program debugging. Such visualizations can be used to depict differences between debugging strategies of expert and novice

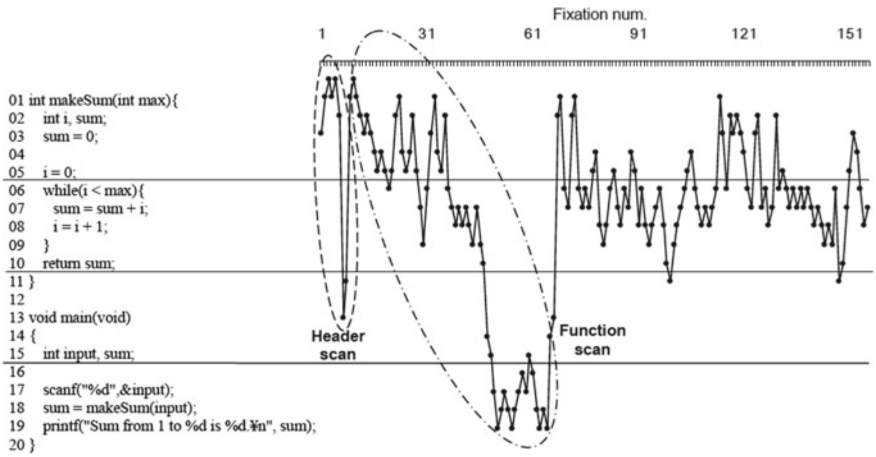


Fig. 24.1 Scanning behavior during program debugging. From Uwano et al. (2006) ©2006 ACM, Inc. Reprinted by permission

programmers and can thus lead to future models of behavior. An example is shown in Fig. 24.1.

24.1.3 Universal Accessibility

Besides pointing in desktop and immersive (VR) displays, the archetypical gaze-based pointing application is eye typing. Eye typing is and has been a useful communication modality for the severely handicapped. Severely handicapped people, possessing an acute need for a communication system, may only be able to control their eyes. Most eye typing systems are implemented by presenting the user with a virtual keyboard, either on a typical computer monitor, or in some cases projected onto a wall. Based on analysis of tracked gaze, the system decides which letter the user is looking at and decides (e.g., by dwell time) whether to type this letter. The system may provide feedback to the user by either visual or auditory means, or a combination of both. Eye tracking systems may be either video-based or based on Electro-OculoGraphic (EOG) potential. An example eye typing interface is shown in Fig. 24.2. Variations of traditional gaze-based pointing may be employed for subjects exhibiting difficulties fixating (e.g., locked-in syndrome). In such cases, if the eyes can only be moved in one direction, the eyes can be used as simple one- or two-way switches and the focus can be shifted from one item to another by using a method

Fig. 24.2 Example of eye typing interface. Courtesy of Prentke Romich Company, Wooster, OH http://www.prentrom.com/access/hm2000.html. Reproduced with permission

known as scanning. Using a combination of scanning and switching, the user can use one switch to change display focus by scanning across the display, then use another switch to indicate selection of the item currently in focus.

Majaranta and Raiha (2002) provide an excellent survey of additional selection and feedback techniques employed in eye typing systems. Majaranta and Raiha have also been instrumental in the creation of COGAIN: Communication by Gaze Interaction,[1] a network of excellence supported by the European Commission's IST sixth framework program. COGAIN integrates cutting-edge expertise on interface technologies for the benefit of users with disabilities. The network aims to gather Europe's leading expertise in eye tracking integration with computers in a research project on assistive technologies for citizens with motor impairments. The COGAIN Web site provides a great deal of information, including (at the time of this writing) three downloadable eye communication systems (Dasher, Gazetalk, and UKO II) and various reports pertaining to the effort.

Although currently the United States lacks a far-reaching universal accessibility program such as COGAIN, some related successes have recently been reported. In particular, Hornof and Cavender (2005) developed *EyeDraw*, a gaze-based drawing program designed to enable children with severe motor impairments to draw with their eyes. The application is based on a simple drawing platform (e.g., draw lines, circles, etc.) but exposes the difficulty inherent with gaze-based control. As with the Midas Touch problem, the critical problem solved by the program is the interpretation of eye movements allowing users to click on buttons, choose drawing start and end points, and save and retrieve drawings. *EyeDraw* is freely available for download.[2] A screenshot with a drawing created solely with eye movements by an *EyeDraw* developer is shown in Fig. 24.3.

24.1.4 Indirect Eye-Based Interaction

Gaze-based communication systems such as those featuring eye typing offer certain (sometimes obvious) advantages but are also problematic. Gaze may provide an often faster pointing modality than a mouse or other pointing device, especially if the targets are sufficiently large. However, gaze is not as accurate as a mouse because the fovea limits the accuracy of the measured point of regard to about 0.5° visual angle. Another significant problem is accuracy of the eye tracker. Following initial calibration, eye tracker accuracy may exhibit significant drift, where the measured point of regard gradually falls off from the actual point of gaze. Together with the Midas Touch problem, drift remains a significant problem for gaze input.

Zhai et al. (1999) take another approach to gaze-based interaction and test the use of gaze as a sort of predictive pointing aid rather than a direct effector of selection. This is a particularly interesting and significant departure from "eye pointing"

[1] See http://www.cogain.org.

[2] See http://www.cs.uoregon.edu/research/cm-hci/EyeDraw/.

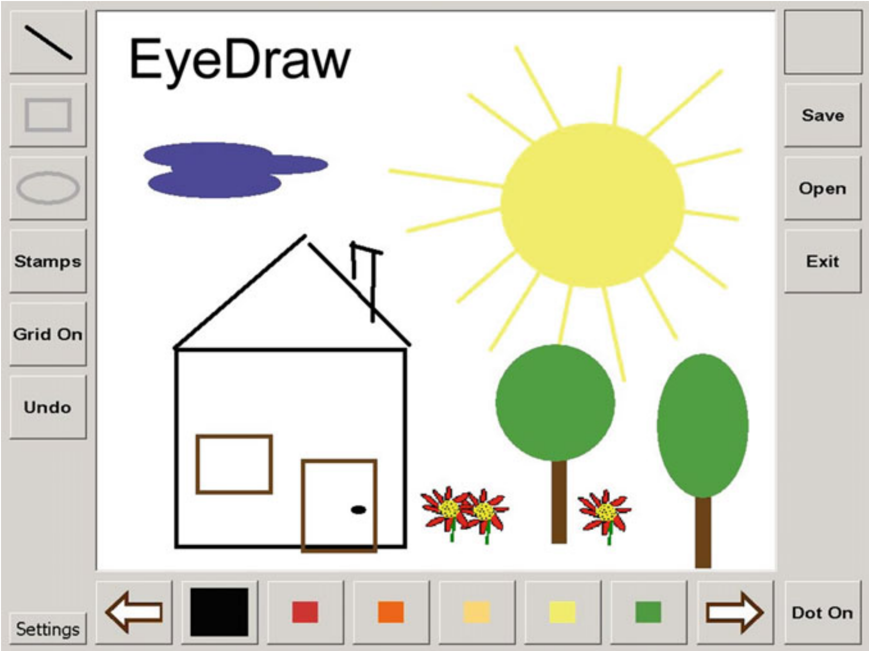


Fig. 24.3 *Sunny Day*. A drawing created solely with eye movements by an *EyeDraw* developer. Publicly available online at http://www.cs.uoregon.edu/research/cm-hci/EyeDraw/ (last accessed 07/10/06)

because this strategy is based on the authors' assertion that loading of the visual perception channel with a motor control task seems fundamentally at odds with users' natural mental model in which the eye searches for, and takes in information, while coordinating with the hand for manipulation of external objects. In their paper on Manual Gaze Input Cascaded (MAGIC) Pointing, Zhai et al. present an acceleration technique where a (two-dimensional) cursor is warped to the vicinity of a fixated target. The acceleration is either immediate, tied to eye movement (liberal MAGIC pointing), or delayed, following the onset of mouse movement (conservative MAGIC pointing). The authors report that although the speed advantage is not obvious over manual (mouse) pointing (subjects tended to perform faster with the liberal method), almost all users subjectively felt faster with either new pointing technique.

Another interesting indirect gaze-based interaction was presented by Santella et al. (2006). Here, eye tracking is used to mark aesthetically important locations in an image. Represented as a gaze-based content map, recorded fixations are then used to automatically crop the image. In a forced-choice ("which one looks better") user comparison, gaze-based croppings were preferred over originals as well as over automatically cropped images. However, professionally hand-cropped images were chosen over any of the other three methods.

24.1.5 Attentive User Interfaces (AUIs)

Enabling a computer application or other physical device awareness of gaze is a topic within the burgeoning area of attentive user interfaces. Vertegaal's (2003) special issue of the *Communications of the ACM* provided several articles falling under this heading. The issue reviews a number of devices that track a user's attention, including eye trackers, attentive phones, and attentive videoconferencing cameras. Developing the notion of AUIs further, Shell et al. (2004) introduced ECSGlasses and EyePliances: devices that used eye contact as an attentional cue to engage in a more sociable process of turn-taking with users. EyePliances relied on an eye detecting camera, the Eye Contact Sensor, for awareness of a user's gaze. Given this cue, the user could simply issue verbal instructions to a gaze-aware device. For example, one could turn on a lamp equipped with an Eye Contact Sensor by simply commanding "On!" Because the lamp could sense it was being looked at, it would "know" to turn itself on.

The requirement that EyePliances needed to be equipped with sensing cameras presented a technical limitation in terms of the camera's field of view. When multiple objects each equipped with an Eye Contact Sensor were placed within 80° of visual angle from each other, an ambiguity arose in determining which object was being fixated. Rather than solving this problem, Smith et al. (2005) removed the camera from the EyePliance and fitted it on a lightweight head-mounted eyepiece. Their revised system for gaze-based *deixis*, or specification of a referent, e.g., "*that* object", by gaze, is termed ViewPointer, and is shown in Fig. 24.4.

Any physical object can be augmented with a small infra-red (IR) tag. To detect eye contact, ViewPointer considers whether the reflection of the IR tag on the cornea appears central to the pupil. If so, the user is looking at the tag. Calibration is not needed because eye movements are not correlated to scene coordinates. In addition to simple eye contact sensing (i.e., a binary decision of whether a device is being looked at), Smith et al. also developed a novel encoding scheme that is used to identify each tag. Other data beyond tag identification can be transmitted visually to the ViewPointer camera at a rate of 14 bits per second (as of that version of the system; it is probable that this rate will increase in future revisions). Information can therefore be transmitted to the ViewPointer wearer as suggested by the usage scenario depicted in Fig. 24.4. In this example, tags are mounted behind the poster in the store window. A URL tag can be transmitted along with tag id information, thereby providing the user with a pointer to relevant Web-based information that ViewPointer could transmit via Bluetooth to a handheld PDA or other browser-enabled device.

24.1.6 Usability

Besides the use of gaze for interactive means, diagnostic eye tracking is gaining acceptance within the HCI and usability communities (particularly practitioners) as

(a) ViewPointer headset

(b) ViewPointer tag and US penny

(c) ViewPointer usage

Fig. 24.4 ViewPointer headset, tag, and usage: a user looks at a poster augmented with an invisible ViewPointer URL tag mounted behind the poster's logo. From Smith et al. (2005) © 2005 ACM, Inc. Reprinted by permission

another means to test usability of an interface. It is believed that eye movements can significantly enhance the observation of users' strategies while using computer interfaces (Goldberg and Kotval 1999). Among various experiments, eye movements have been used to evaluate the grouping of tool icons, compare gaze-based and mouse interaction techniques (Sibert and Jacob 2000), evaluate the organization of click-down menus, and to test the organization of Web pages.

In a usability study of Web pages, Goldberg et al. (2002) derive specific recommendations for a prototype Web interface tool. The authors discuss gaze-based evaluation of Web pages in which the system permits free navigation across multiple Web pages. This is a significant advancement for usability studies of Web browsers inasmuch as prior to this study recording of gaze over multiple Web pages has been difficult due to the synchronization of gaze over windows that scroll or hide from view. The authors describe their collection of the following dependent measures.

1. User actions such as key presses and mouse button clicks
2. Context-free eye movement measures such as fixations and dwell times
3. Context-sensitive eye movements such as dwell times within regions of interest

The key questions examined are whether eye movements are related to user actions, is navigation biased toward horizontal or vertical navigation, and whether there is any particular order to Web page scanning. Following task-level, screen-level, and object-level analyses, the authors report that users exhibit a preference for horizontal search across columns rather than searching within a column. Furthermore, recommendations are made for the left-hand placement of Web "portlets" requiring visibility, with the two most important portlets placed on top of the Web page.

A good deal many more gaze-based usability studies have been conducted. The above are but two examples, with the first being perhaps more relevant to user modeling than usability practice. However, since Karn et al. (1999) well-attended SGICHI 99 workshop, eye tracking appears to have gathered some momentum among usability practitioners. Following Jacob and Karn (2003) review of eye movement metrics, Bojko (2005) presented useful eye tracking pointers garnered from practical experience. Bojko's study is mentioned in Chap. 18 where Guan et al. (2006) results on Retrospective Think-Aloud are also discussed. Results from Bojko et al. (2005) exemplary drug label experiment are covered in Chap. 23. A host of other practical issues were discussed in two more recent workshops aimed at practitioners, one again at SIGCHI 06 and another at UPA 06 (Renshaw et al. 2006; Webb and Renshaw 2006). The recent findings of the "F" and "golden triangle" Web search scan patterns (attributed to Nielson Norman Group (2006) and Eyetools et al. (2006), respectively, see Chap. 23) suggest that eye tracking will continue to play an increasingly important role in usability investigations.

24.1.7 Collaborative Systems

Apart from interactive or diagnostic uses of eye movements, gaze can also be utilized to aid multiparty communication in collaborative systems. In the GAZE Groupware system, an LC Technologies eye tracker is used to convey gaze direction in a multiparty teleconferencing and document sharing system, providing a solution to two problems in multiparty mediated communication and collaboration: knowing who is talking to whom, and who is talking about what (Vertegaal 1999). The system displays 2D images of remotely located participants in a VRML virtual world. These images rotate to depict gaze direction alleviating the problem of turn-taking in multiparty communication systems. Furthermore, a gaze-directed "lightspot" is shown over a shared document indicating the users' fixated regions and thereby providing a deictic ("look at this") reference. The system display is shown in Fig. 24.5, with the the system interface shown in Fig. 24.6. For further information, see http://www.cs.queensu.ca/home/roel/gaze/home.html.

24.2 Gaze-Contingent Displays

In general, eye-based interactive applications can be thought of as selective, because gaze is used to select or point to some aspect of the display, whether it is two-dimensional (e.g., desktop), collaborative, or immersive (such as a virtual

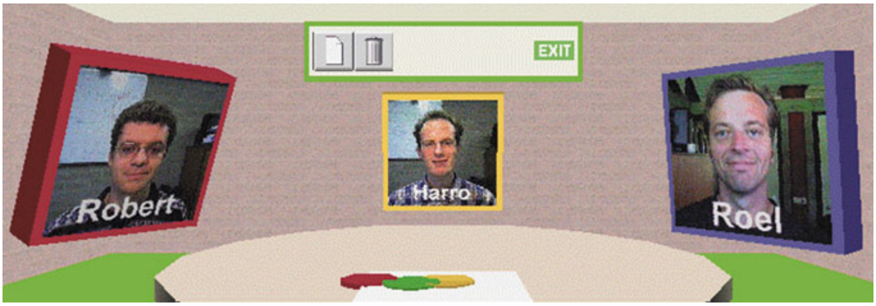
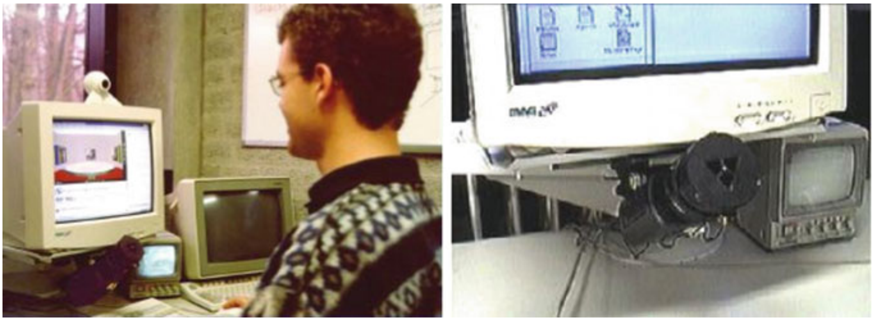


Fig. 24.5 GAZE Groupware display. Courtesy of Roel Vertegaal



(a) User interface. (b) Eye tracking optics.

Fig. 24.6 GAZE Groupware interface. Courtesy of Roel Vertegaal

environment). Mixing both directly interactive and indirectly "passive" usage styles of gaze are gaze-contingent displays. Here, gaze is used not so much as a pointing device, but rather as a passive indicator of gaze. Given the user's point of regard, a system can tailor the display so that the most informative details of the display are generated at the point of gaze, and degraded in some way in the periphery. The purpose of these displays is usually to minimize bandwidth requirements, as in video telephony applications, or in graphical applications where complex data sets cannot be fully displayed in real-time. Two main types of gaze-contingent applications are discussed: screen-based and model-based. The former deals with image (pixel) manipulation, and the latter is concerned with the manipulation of graphical objects or models. Both systems are generally investigated by researchers studying Computer Graphics (CG) and Virtual Reality (VR).

Human visual perception of digital imagery is an important contributing factor to the design of perceptually based image and video display systems. Human observers have been used in various facets of digital display design, ranging from estimation of corrective display functions (e.g., gamma function) dependent on models of human color and luminance perception, color spaces (e.g., CIE Lab color space), and image and video codecs. JPEG and MPEG both use quantization tables based on the notion of Just Perceptible Differences to quantize colors of perceptually similar hue (Wallace 1991).

The idea of gaze-contingent displays is not new and dates back to early military applications (Kocian 1987; Longridge et al. 1989). 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 that 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.

Various gaze-contingent approaches have been proposed for foveal Region Of Interest (ROI)-based image and video coding (Stelmach and Tam 1994; Nguyen et al. 1994; Kortum and Geisler 1996; Tsumura et al. 1996). Often, however, these studies are based on automatically located image regions, which may or may not correspond to foveally viewed segments of the scene. That is, these studies do not necessarily employ an eye tracker to verify the ROI-based coding schemes. Instead, a figure-ground assumption is often used to argue for more or less obvious foveal candidates in the scene. This is a research area where either diagnostic eye movement studies can be used to corroborate the figure-ground assumption, or gaze may be used directly to display high-resolution ROIs at the point of regard in real-time (as in eye-based teleconferencing systems).

Instead of assuming a feature-based approach to foveal (high-resolution) encoding, an eye tracker can be used to directly establish the foveal ROI and a suitable image degradation scheme may be employed to render detail at the point of regard. This motivated research into finding a suitable image degradation scheme that would match foveal acuity.

24.2.1 Screen-Based Displays

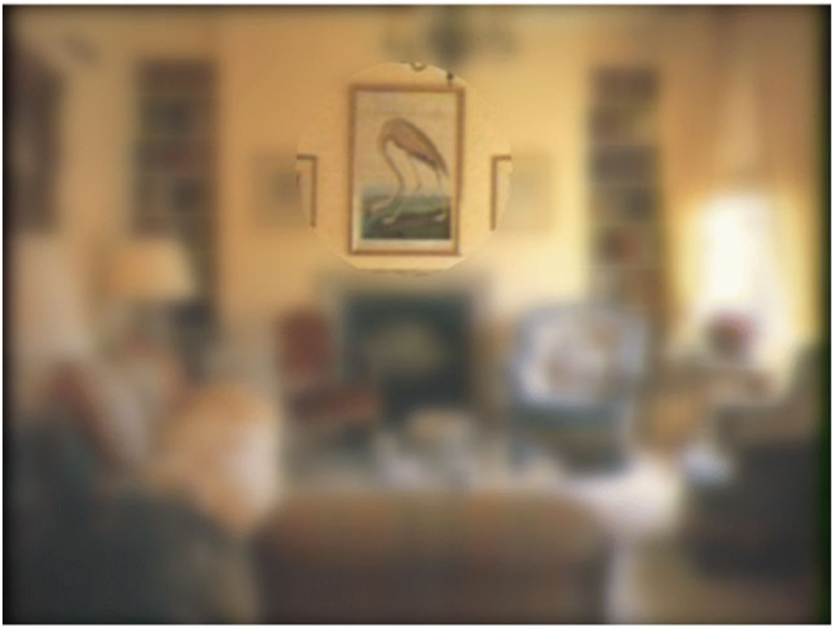
When evaluating Gaze-Contingent Displays (GCDs), it is often necessary to distinguish between two main types of causalities: those affecting perception and those affecting performance. As a general rule, perception is more sensitive than performance. That is, it may be possible to degrade a display to a quite noticeable effect without necessarily degrading performance. In either case, one of the main difficulties that must be addressed is the latency of the system. Without predictive capabilities, most gaze-contingent displays will lag behind the user's gaze somewhat, usually by a constant amount of time proportional to both measurement of gaze (which may take up to the time it takes to display a single frame of video, typically 16 ms for a 60 Hz video-based tracker), and the subsequent time it takes to refresh the gaze-contingent display (this may take another 33 ms for a system with update rate of 30 frames per second).

Loschky and McConkie (2000) conducted an experiment on a gaze-contingent display investigating spatial, resolutional, and temporal parameters affecting perception and performance. Two key issues addressed by the authors 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. An example of Loschky and McConkie's GCD is shown in Fig. 24.7a. In one facet of the experiment, it was found that for an image change to go undetected, it must be started within 5 ms after the end of the eye movement. Detection likelihood rose quickly beyond that point. In another facet of the study concerning detection of peripheral degradation, results showed that the least peripheral degradation (inclusion of four of four possible levels) 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. In the end it was found that the generation of an imperceptible GCD was quite difficult in comparison to the generation of a GCD that does not deteriorate performance. Although greater delays (e.g., 15 ms) and greater degradation (inclusion of only three of four possible levels) 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.

Parkhurst et al. (2000) investigated behavioral effects of a two-region gaze-contingent display. A central high-resolution region, varying from 1 to 15 degrees, was presented at the instantaneous center of gaze during a visual search task. An example of Parkhurst et al. display is shown in Fig. 24.7b. Measures of reaction time, accuracy, and fixation durations were obtained during a visual search task. The authors' 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 region 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°.

For screen-based VR rendering the work of Watson et al. (1997) is particularly relevant. The authors studied the effects of Level Of Detail (LOD) peripheral

(a) From Loschky and McConkie (2000) © 2000 ACM, Inc. Reprinted by permission



(b) From Parkhurst et al. (2000) © 2000 ACM, Inc. Reprinted by permission

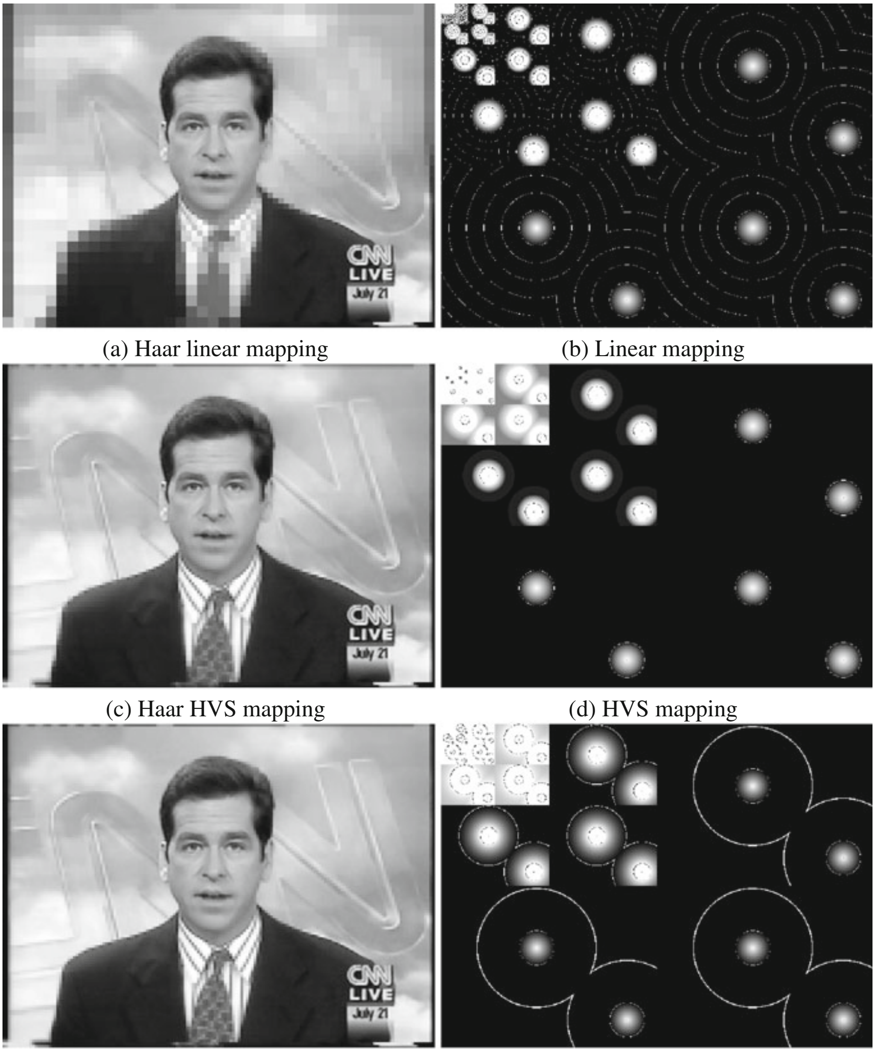
Fig. 24.7 Example gaze-contingent displays degradation on visual search performance. Both spatial and chrominance detail degradation effects were evaluated in Head-Mounted Displays (HMDs). To sustain acceptable frame rates, two polygons were texture mapped in real-time to generate a high-resolution inset within a low-resolution display field. The authors suggested that visual spatial and chrominance complexity can be reduced by almost half without degrading performance. More recently, Watson et al. (2004) used the same head-mounted display (but not head-tracked this time) to gain insights into peripheral LOD control beyond the perceptual threshold.

In an approach similar to Watson et al., Reddy (1998) used a view-dependent LOD technique to evaluate both perceptual effects and system performance gains. The author reported a perceptually modulated LOD system which affords a factor of 4.5 improvement in frame rate.

Most of the above approaches result in a bi-resolution GCD, with the demarcation between the foveal disk region and peripheral resolution often purposefully made visible (thus without any inter-LOD blurring or averaging). Geisler and Perry (2002) proposed a method to generate completely arbitrary variable resolution displays. Their display depends on pyramidal preprocessing of the images prior to display (Geisler and Perry 1998) (see Burt and Adelson 1983 for a detailed description of multiresolution pyramids with spatial filtering). Geisler and Perry *Space Variant Imaging* software produces smooth nearly artifact-free images at high frame rates, but is limited to manipulation of spatial resolution. The software implementing this method on Windows platforms is freely available on-line.[3] Geisler and Perry work is particularly significant for its separation of resolution degradation from image source. In image compositing parlance, this *switchmatte* operation makes gaze-contingent rendering immediately obvious. Simply preserve high-resolution pixels only at matte locations with $\alpha = 1$ and map pixels at lower matte luminance levels to lower-resolution pixels (e.g., from a bank of preprocessed images).

Because pyramidal reconstruction schemes draw pixel data from multiple levels of resolution, pyramidal image synthesis provides a smoothly degraded, convincing visualization of the human visual system, termed *foveation*. A particularly popular pyrmadical approach relies on image decomposition via the Discrete Wavelet Transform (DWT) with selective coefficient scaling and decimation prior to reconstruction (Chang et al. 2000; Duchowski 2000). Using the DWT for smooth image resolution degradation, images demonstrating three acuity mapping functions are shown in Fig. 24.8. For demonstration purposes, the *CNN* image was processed with an artificially placed ROI over the anchor's right eye and another over the "timebox" found in the bottom-right corner of the image. Haar wavelets were used to accentuate the visibility of resolution bands (see Sect. 4.2). Figure 24.8b, d, and f show the extent of wavelet coefficient scaling in frequency space. The middle row shows a reconstructed image where resolution drops off smoothly, matching visual acuity for a particular screen display at a particular viewing distance. Provided appropriate wavelet filters can be found, reconstruction exactly matches linear mipmapping.

[3] http://fi.cvis.psy.utexas.edu.



(a) Haar linear mapping (b) Linear mapping

(c) Haar HVS mapping (d) HVS mapping

(e) Haar nonlinear mapping (f) Nonlinear mapping

Fig. 24.8 Image reconstruction and wavelet coefficient resolution mapping (assuming 50 dpi screen resolution). Reprinted from Duchowski (2000) with permission © 2000 IEEE

Recently, Cöltekin (2006) used a pyramid-based LOD management method for close-range stereo photogrammetric rendering. Meanwhile, Böhme (2006) GCD goes beyond spatial resolution degradation by degrading temporal content as well as spatial content. Spatiotemporal degradation extends Geisler and Perry pyramidal preprocessing approach into the temporal dimension.

Although recent GCD implementations have yielded new insights into perception, most of these approaches still rely on somewhat restrictive computational approaches, i.e., either software-based pyramidal image reconstruction (as in Böhme spatiotemporal video degradation) or a texture-mapped rendering with a limited number of textures (as in Watson et al. dual viewport composition). The limitation of these techniques surfaces either in their limited speed or display characteristics. Böhme claim 30 frames per second performance but have previously reported an average system latency of 60 ms (Dorr et al. 2005). Watson et al. dual viewports limited the display to two distinct regions. It is unlikely that this approach lends itself to the simulation of arbitrary visual fields. Indeed, Reddy (2001) noted that practically all perceptually based work (up to that point) had used a small set of presimplified versions of an object from which to choose to render in a view-dependent manner. It appears that this is still the predominant approach without the apparent employment of the GPU for image synthesis. Reddy *Percept* visualization performs a per-pixel calculation of the pixel's spatial frequency based on angular velocity and eccentricity whereas Cöltekin *Foveaglyph* builds a pyramid of scaled images.

Due to recent advancements in computer hardware, gaze-contingent imaging research has appeared where image processing operations are performed in real-time, either by dedicated image processing hardware, or by more general-purpose graphics engines. In a recent example of dedicated hardware-accelerated eye-movement controlled image coding, Bergström (2003) used a DCT-based image codec to achieve real-time image compression and display.

A GPU-based gaze-contingent display is now available[4] with minimal progamming effort. A short GLSL program allows simulation of arbitrary visual fields, as inspired by Geisler and Perry (2002), but degraded chromatically as well as spatially. The approach is a natural extension of Nikolov et al. (2004) and Duchowski (2004) independent introduction of multitexturing approaches.

24.2.2 Model-Based Graphical Displays

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 (1976), is now standard practice, particularly in real-time applications such as VR (Vince 1995). Clarke's original criterion 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 precomputed fine-to-coarse hierarchies of an object. This leads to uniform, or *isotropic*, object resolution degradation.

[4] http://andrewd.ces.clemson.edu/gcd/.

Ohshima et al. (1996) proposed a gaze-contingent model-based adaptive rendering scheme where three visual characteristics were considered: central/peripheral vision, kinetic vision, and fusional vision. 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. Because 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 (Zorin and Schröder 2000). 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 (Lindstrom et al. 1996; MacCracken and Joy 1996; Hoppe 1997; Zorin et al. 1997; Schmalstieg and Schaufler 1997). 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 nonisotropically.

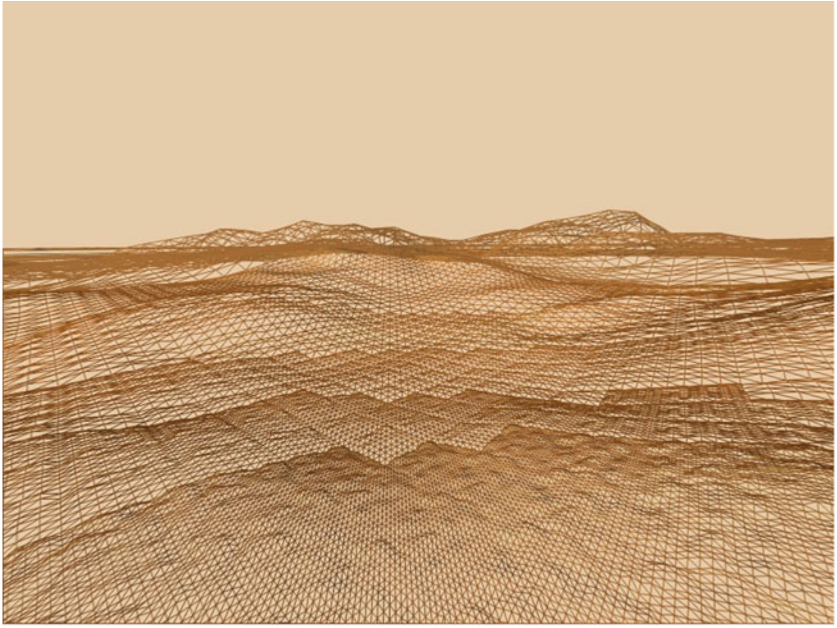
An early example of a nonisotropic model-based gaze-contingent system, where gaze direction is directly applied to the rendering algorithm, was presented by Levoy and Whitaker (1990). The authors' spatially adaptive near-real-time ray tracer for volume data displayed an eye-slaved ROI by modulating both 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 convolution 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 in. TV monitor. A chin rest and immobilization strap were used to eliminate the need for head tracking.

For environments containing significant topological detail, such as virtual terrains, 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 (Danforth et al. 2000). Recent work in this area has been extensive. Particularly impressive is Hoppe (1998) view-dependent progressive mesh framework, where spatial continuity is maintained through structure design, and temporal continuity is maintained by geomorphs.

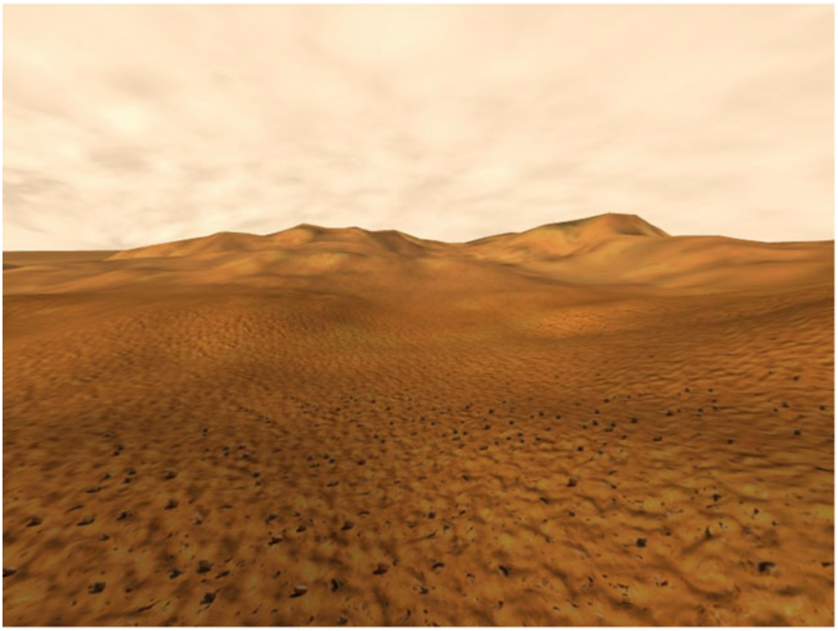
Danforth et al. (2000) used an eye tracker as an indicator of gaze in a gaze-contingent multiresolution terrain navigation environment. A surface, represented as a quadrilateral mesh, was divided into fixed-size (number of vertices) subblocks, allowing rendering for variable LOD on a per-subblock basis. From a fully detailed surface, lower levels of resolution were constructed by removing half of the vertices in each direction and assigning new vertex values. The new values were averages of the higher resolution values. Resolution level was chosen per sub-block, based on viewer distance. The resolution level was not discrete; it was interpolated between the precomputed discrete levels to avoid "popping" effects. The terrain, prior to gaze-contingent alteration, is shown in Fig. 24.9. Rocks in the terrain are rendered by billboarding; i.e., images of rocks from the Pathfinder mission to Mars were rendered onto 2D transparent planes that rotate to maintain an orientation orthogonal to the viewer. Two views of the gaze-contingent environment (shown rendered and in wireframe) are seen in Figs. 24.10 and 24.11. To exaggerate the gaze-contingent effect, in this environment, fractal mountains appear and disappear from view, based on direction of gaze. Notice also, in Fig. 24.10a, b, the increased resolution (number of quads) below the gaze vector. The images in the figure are snapshots of the scene images generated by the eye tracker; i.e., what is seen by the operator (the point of regard crosshair, coordinates, and video frame timecode) is not seen by the viewer immersed in the environment.

More recent work on gaze-contingent LOD modeling has been carried out by Luebke and Erikson (1997). The authors present a view-dependent LOD technique suitable for gaze-contingent rendering. Although simplification of individual geometric objects is discussed in their work, it appears the strategy is ultimately directed toward solving the interactive "walkthrough" problem (Funkhouser and Séquin 1993). In this application, the view-dependent LOD technique seems more suitable to the (possibly) gaze-contingent rendering of an entire scene or environment. Recently, the authors have developed a gaze-directed LOD technique to facilitate the gaze-contingent display of geometric objects (Luebke et al. 2000). 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. This work shows the feasibility of employing an eye tracker, however the implementation framework used by the authors lacked a head tracker and required a chin rest to ensure tracker accuracy.

A new object-based LOD method has been developed by Murphy and Duchowski (2001). The technique is similar to Luebke and Erikson and to Ohshima et al., where objects are modeled for gaze-contingent viewing. Unlike the approach of Ohshima et al., resolution degradation is applied nonisotropically; i.e., objects are not necessarily degraded uniformly. The spatial degradation function for LOD selection differs significantly from the area-based criterion originally proposed by Clarke. Instead of evaluating the screen coverage of the projected object, the degradation function is based on the evaluation of visual angle in world coordinates. System performance measurements are reported, obtained from experiments using a binocular eye tracker built into an HMD. 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

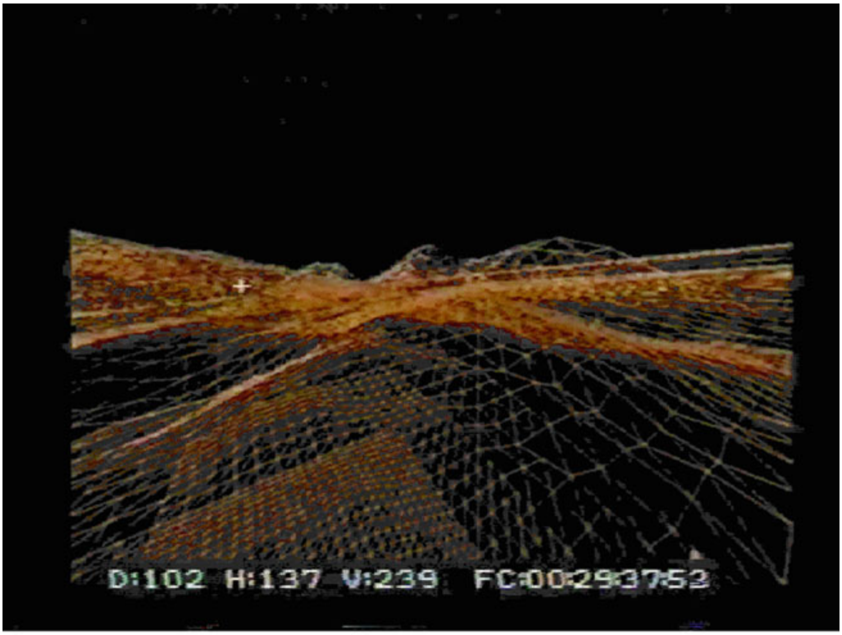


(a) Wireframe

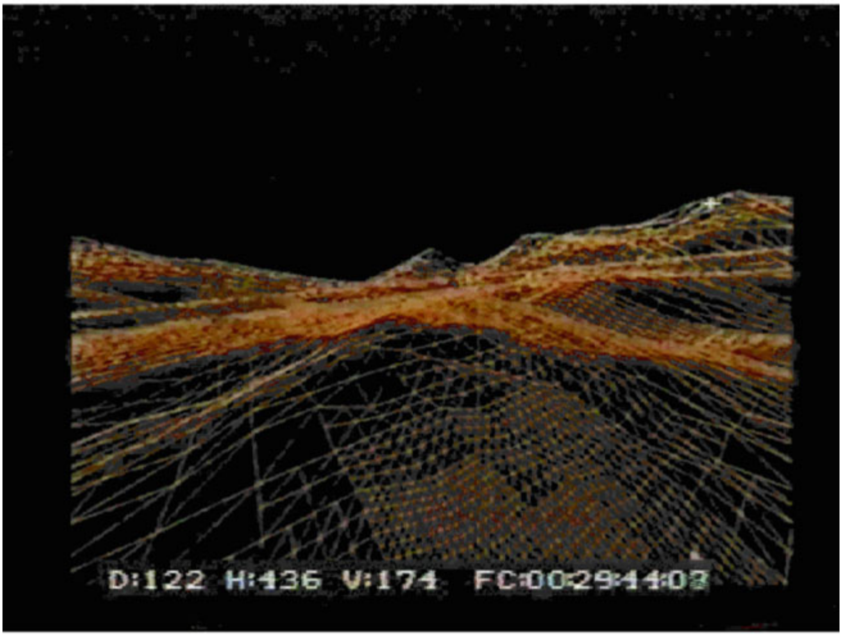


(b) Rendered

Fig. 24.9 Fractal terrain for gaze-contingent virtual environment. Courtesy of Bob Danforth



(a) Looking left



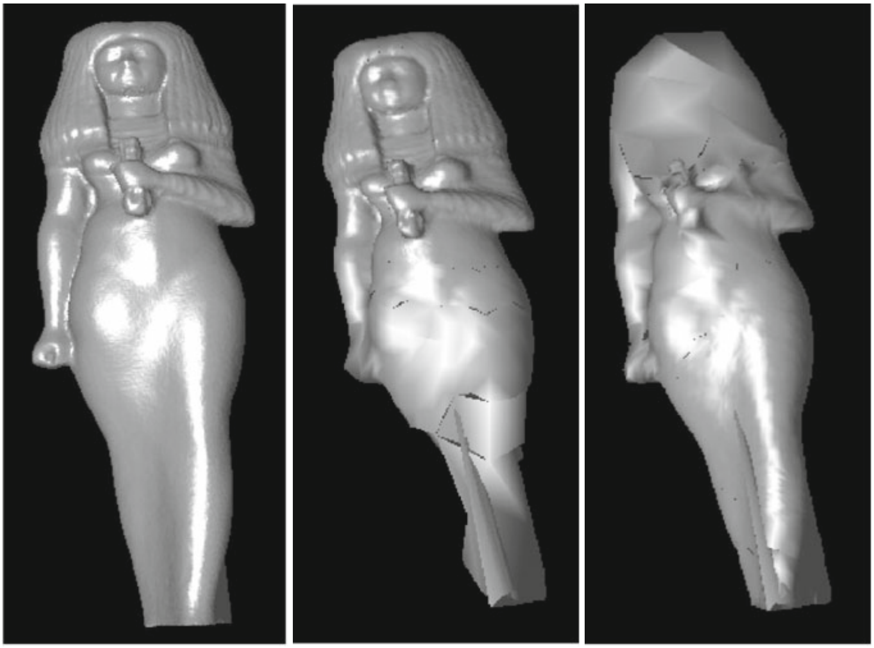
(b) Looking right

Fig. 24.10 Fractal terrain: gaze-contingent rendering (wireframe)

(a) Looking left

(b) Looking right

Fig. 24.11 Fractal terrain: gaze-contingent rendering



(a) Original image (b) Looking at head (c) Looking at feet

Fig. 24.12 Gaze-contingent viewing of *Isis* model. Courtesy of Hunter Murphy

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 tenfold average frame rate improvement during gaze-contingent viewing. Two frames during gaze-contingent viewing of one geometric model are shown in Fig. 24.12.

Another interesting approach to gaze-contingent modeling for real-time graphics rendering was taken by O'Sullivan and Dingliana (2001) and O'Sullivan et al. (2002). Instead of degrading the resolution of peripherally located geometric objects, O'Sullivan and Dingliana (2001) considered a degradable collision handling mechanism to effectively limit object collision resolution outside a foveal Region Of Interest (ROI). When the viewer is looking directly at a collision, it is given higher (computational) priority than collisions occurring in the periphery. Object collisions given higher priority are allocated more processing time so that the contact model and resulting response are more believable. The calculated collision response accuracy (i.e., the precision of the calculated collision point between two rigid objects) is directly dependent on the level of collision detection detail. A lower level of detail results in less physically accurate physics and objects not touching when they bounce, leaving a potentially perceivable gap.

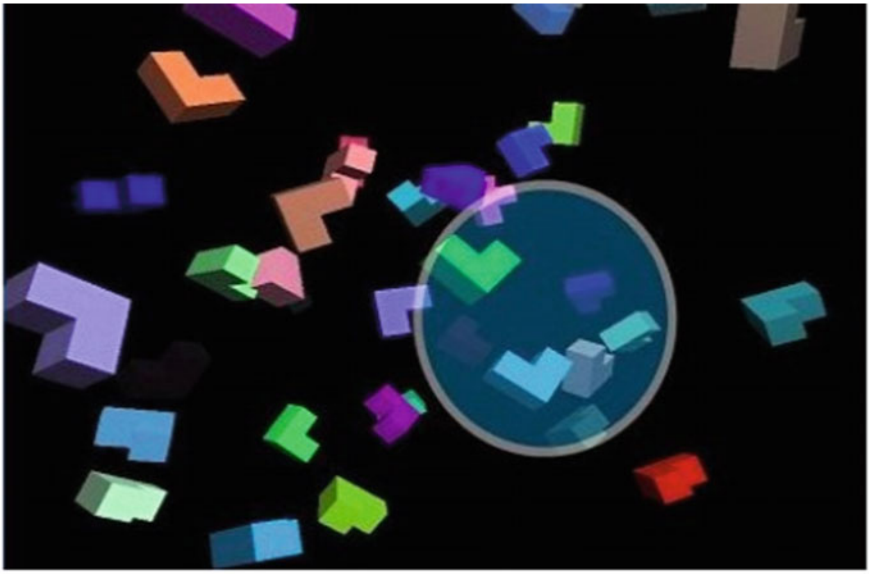
O'Sullivan and Dingliana (2001) describe psychophysical experiments to measure the perceptibility of coarsely simulated collisions in the periphery. The authors tested effects of factors such as eccentricity, separation, presence and number of similar and dissimilar distractors, causality, and physics on participants' perception of collisions. To test viewers' sensitivity to collision resolution (size of gap between colliding objects) at eccentricity, O'Sullivan and Dingliana tested whether the ability of viewers to detect anomalous collisions, in this case colliding objects that do not touch each other but leave a gap, decreases with increasing eccentricity of the collision point. Collisions were presented at five eccentricities of 1.4°, 2.9°, 4.3°, 5.7°, and 7.2° visual angle. The authors note a significant fall-off in detection accuracy with eccentricity (at about 4° visual angle).

Based on their previous psychophysical findings, O'Sullivan et al. (2002) developed a gaze-contingent collision handling system. Two variants of the gaze-contingent system were compared, each containing a high-priority ROI wherein collisions were processed at higher resolution than outside the ROI. In the tracked case, the high-priority ROI was synchronized to the viewer's tracked gaze position with an SMI EyeLink eye tracker, and in the random case, the ROI position was determined randomly every five frames. An example of the graphics simulation with highlighted ROI and subject wearing an eye tracker is shown in Fig. 24.13. O'Sullivan et al. report an overall improvement in the perception of the tracked simulation.

24.3 Summary and Further Reading

Several classes of eye tracking applications were presented falling generally within the domain of computer wcience and human–computer interaction. HCI-related studies have included some of the first well-known adaptations of eye trackers to computer-based systems. This trend will probably continue for some time. Interactive eye tracking systems including those featured in Computer-Supported Collaborative Work (CSCW) will most likely continue to be explored for directly interactive support (e.g., gaze pointing) as well as for indirect interaction (e.g., gaze-assisted pointing). However, the recent interest in diagnostic uses of eye trackers, especially in usability studies, is also expected to flourish.

For an excellent review of Gaze-Contingent Multi-Resolution Displays, or GCMRDs, see Reingold et al. (2002) as well as Parkhurst and Niebur (2002). The remainder of gaze-contingent work described in this chapter mostly emanates from research in computer graphics. Gaze-contingent displays described here are usually interactive real-time examples of adoption of eye trackers into graphical displays. See O'Sullivan et al. (2002) for a review of eye tracking work in interactive graphics. Here too, eye trackers are becoming noticed for their diagnostic contributions. The recent ACM SIGGRAPH Campfire on Perceptually Adaptive Graphics (McNamara and O'Sullivan 2001) suggests that general perceptual issues are becoming particularly important in computer graphics research. Traditional graphics algorithms (e.g., ray-tracing, radiosity) are beginning to include perceptually based improvements to further speed up processing time. Eye trackers will certainly be able to contribute in a diagnostic capacity to improve the fidelity of perceptually enhanced graphical imagery.



(a) Important collisions (e.g., those close to the viewer's fixation position) are processed first

(b) An eye-tracker is used to determine the viewer's point of fixation

Fig. 24.13 Gaze-contingent collision modeling. Images courtesy of C. O'Sullivan and J. Dingliana

For further reading, the two best sources of current research in HCI and computer graphics are the annual ACM SIGCHI and ACM SIGGRAPH conferences and the related ACM journal articles in the *Transactions on Computer–Human Interaction* and the *Transactions on Graphics*. Work specifically related to virtual reality can be found in the ACM VRST and IEEE VR conferences as well as in the journal *Presence*.

Work related to perception can be found in the recently formed *Transactions on Applied Perception (TAP)*, *Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, and the proceedings of the Symposium on Applied Perception in Graphics and Visualization (APGV).