Real-Time Simulation of Visual Defects with Gaze-Contingent Display

Margarita VINNIKOV[1], Robert S. ALLISON[1], Dominik SWIERAD[1]

[1] Center for Vision Research, Department of Computer Science and Engineering, York University, Toronto, Ontario, Canada

Abstract

Effective management and treatment of glaucoma and other visual diseases depend on early diagnosis. However, early symptoms of glaucoma often go unnoticed until a significant portion of the visual field is lost. The ability to simulate the visual consequences of the disease offers potential benefits for patients and clinical education as well as for public awareness of its signs and symptoms. Experiments using simulated visual field defects could identify changes in behaviour, for example during driving, that one uses to compensate at the early stages of the disease's development. Furthermore, by understanding how visual field defects affect performance of visual tasks, we can help develop new strategies to cope with other devastating diseases such as macular degeneration. A Gaze-Contingent Display (GCD) system was developed to simulate an arbitrary visual field in a virtual environment. The system can estimate real-time gaze direction and eye position in earth-fixed coordinates during relatively large head movement, and thus it can be used in immersive projection based VE systems like the CAVETM. Arbitrary visual fields are simulated via OpenGL and Shading Language capabilities and techniques that are supported by the GPU, thus enabling fast performance in real time. In order to simulate realistic visual defects, the system performs multiple image processing operations including change in acuity, brightness, color, glare and image distortion. The final component of the system simulates different virtual scenes that the participant can navigate through and explore. As a result, this system creates an experimental environment to study the effects of low vision on everyday tasks such as driving and navigation.

Keywords: Head-Eye tracking System, Gaze Contingent Display, Image processing, Visual fields, Low vision, Eye disease, Foveated imaging, Variable resolution image

1 Introduction

Tracking of the eye movements is widely used by many disciplines such as neurology, ophthalmology and psychology as a diagnostic tool [Allison et al, 1997; Leigh and Zee 1991]. In most cases the tracking is passive and is used for pure information retrieval. Another class of applications known as Gaze Contingent Displays (GCD) capitalizes on close coupling of eye movements and vision by having the display of information tied to the user's gaze direction.

As a research tool, GCD systems offer the most flexibility when incorporated into immersive virtual environment technology. An immersive virtual environment (VE), also known as virtual reality (VR), is a synthetic environment that the user can manipulate and interact with. An immersive VE allows for a wide range of natural interactions and can be used to perform studies of complex behaviours with greater repeatability, reliability and experimental control than in complex natural environments. Typically the user's body is tracked to increase the sense of immersion and to facilitate a degree of natural interaction.

2 Visual Field simulation

Early GCD systems were simple and used primarily for the study of visual phenomena such as perceptual fading of stable retinal images and saccadic suppression [Bridgeman at al. 1975; Dodge 1900]. In information display and simulation, GCD algorithms have been used for a variety of purposes. One of the most basic techniques capitalizes on the non-uniform distribution of photoreceptors on the human retina by concentrating display resolution or computational resources at the point of fixation. Initially proposed in the 1970's, the technique was not introduced in commercial systems until the early 1990s, most notably in the CAE FOHMD which included a high-resolution electronic display inset slaved to the user's eye position [Robinson, Thomas and Wetzel 1989]. Advances in multi-resolution texture mapping and other graphics algorithms and hardware facilitated the development of gaze-contingent rendering of multi-resolution displays [Duchowski 2002]. Similarly, Geisler and Perry [2002] have proposed using GCD algorithms to perform image compression and to create educational simulations to make people aware of different signs of vision loss. Nikolov et al. [2004] developed a gaze-contingent application using OpenGL and texture mapping techniques to highlight points of focal interest on the screen.

In this paper we will describe several image processing algorithms that can be used to create visual defects in a real time as well as in our virtual environment setup.

3 Point of Regard and Line of Sight

The point of one's fixation is referred to as the point of regard (POR). The direction of gaze from the center of the eye to the point of regard is referred to as a line of sight (LOS) [figure 1]. The point of regard and the line of sight can be estimated from measurements of eye and head positions and orientations. The eye position needs to be converted to the world coordinate frame (*WCF*) by combining head tracker estimates with the spatial relationship between the head tracker and the eye. A set of homogeneous transformations must be performed to convert the line of sight from the eye coordinate frame (*ECF*) to the world

ETRA 2008, Savannah, Georgia, March 26–28, 2008.
© 2008 ACM 978-1-59593-982-1/08/0003 $5.00

coordinate frame (*WCF)*. Satisfactory performance depends on careful calibration, particularly when the display and tracking systems are decoupled.(For example with an earth-fixed display and a head-mounted eye tracker). For details on how this is achieved in our system please see Huang [2004].

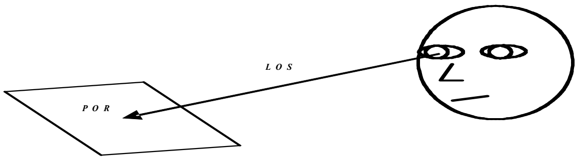

Figure 1: Line of sight (LOS) and point of regard (POR)

4 Image Processing

Many eye diseases and disorders are characterized by visual defects at specific retinal locations. The visual field is a map of visual capability (for instance acuity) as a function of retinal location. Visual fields can be simulated by GCD due to the isomorphism that is established between the display and the user's retina. There are two types of GCDs: Model-based and Screen-based algorithms. Model-based algorithms are 3D computer graphic techniques that manipulate and render modeled objects to various levels of detail based upon current gaze direction. The challenges in implementing these algorithms lie in performing calculations in 3D and accommodating for partial visual changes within an object, limiting their applicability to the simulation of visual fields. Screen-based applications perform pixel manipulation directly on the 2D screen based on current gaze direction. This could be done optically or electronically as in the early GCDs, but today, it is typically a video processing step. Although the screen-based approach is a natural match for visual field simulation (which intends to create a controlled 2D retinal image), most screen–based applications work with video streams and usually are not applied to virtual 3D environments. Thus, the use of video streams limits the experimental conditions by having a predefined view and motion, and they do not reflect real-world situations [Duchowski 2003]. Flexible virtual environments are required in experimental situations where naturalistic behaviour is examined. There is a need for visual field simulation that will combine the advantages of both approaches: the image-centricity and simplicity of screen-based approach, combined with the ability to interact with 3D models, and leveraging of the capabilities of 3D graphical processor hardware of the model-based approach.

Geisler and Perry [2002] described an algorithm that can simulate arbitrary visual fields for both normal individuals and low-vision patients. Although their algorithm permitted updating the images during run time, it was limited to 2D video streams only. Our system simulates 3D virtual environments utilizing OpenGL hardware acceleration but instead of using Model-based algorithms, it implements Screen-based algorithms at a graphic card level via Shading Language techniques. Systems based on shaders have not been previously reported. However, while this paper was in initial review, a thorough survey paper by Duchowski and Coltekin [2007] was published that discussed the benefits of shading language techniques, which described an approach to shader-based GCD that is similar to ours, and it also presented snippets of suitable shader language code. Shading language techniques for GCD systems offer increased flexibility and performance benefits for designers. They are fast and efficient because they do not have to handle 3D calculations and do not have to change the shading and appearance of each individual object in a scene, but rather work with incoming vertex or fragment values and their associated data at the GPU level. Moreover, a shading language approach allows delegating picture analysis and calculations to the GPU thus freeing the CPU for other operations.

We have identified several possible image processing operations that affect spatial resolution, saturation, brightness, contrast, spatial distortion and glare in order to simulate a wide range of visual defects when combined together. Most of the image processing operations are handled in a single rendering pass, while the passes are combined to simulate complex defects. This makes the system perform faster, and hence, preserves the sense of immersion that is important for experiments in virtual environments. Furthermore, the simulation of an arbitrary visual field is done by using OpenGL and Shading Language capabilities and techniques that are supported by the GPU (nVidia), thus supporting more efficient real time performance.

During each frame the following actions are executed:

A high-resolution image is extracted from the virtual camera. This is achieved by using a frame buffer object (FBO) that is an OpenGL extension that supports flexible off-screen rendering. Most importantly this does not require context switching and is efficient due to the fact that resources are shared within the same context. Thus, it is much faster than other known methods.

For each image processing operation a different visual field map is used. A visual field map is an RGBa texture and therefore stores up to 4 channels of information per location. As a result, if the texture is large enough, it can provide information for each pixel on the screen. For spatial resolution processing, for example, it is enough to have one channel of information since the visual field (resolution) map in this case describes the resolution levels across retinal eccentricities. For distortion processing, on another hand, all three channels are required to provide a normal for each pixel.

Based on the POR and LOS estimations, as described in Huang [2004], we transform the visual field map such that the area on the visual field map that corresponds to the fovea is centered on the fovea of the participant's eye. Generally this is a non-linear transformation to project the map from the spherical retina to the planar screen [Fortenbaugh 2007]. Once again we are using the FBO capabilities and therefore there is no slowdown in performance despite the fact that the visual field map is regenerated for every visual artifact.

Both the visual field maps and the image from the virtual camera are then passed to GPU for image processing. The resulting image is then either rendered back to the FBO in order to perform additional image processing or directly rendered to the screen if it is the final image.

In the next sections we will briefly describe specifics for each Image-processing algorithm.

4.1 Spatial Resolution Algorithm

The algorithm we have implemented for spatial resolution is a modified version of the algorithm proposed by Geisler and Perry [1998]. Very recently, Fortenbaugh at al. [2007] implemented this algorithm using a binocular HMD system to study the roles of the visual fields during navigation through a 3D virtual environment. Their system is designed to model changes in acuity and does not handle any other type of visual deficit that we consider. Their approach is similar to ours in that they use the GPU to do most of the graphical processing. However their algorithm processes 8 images for each frame. In contrast to our approach, the entire blur simulation is performed in a single rendering pass. The original algorithm requires creating a multiresolution pyramid starting from a source image. This can be achieved by creating mipmaps in OpenGL. FBO processing supports very fast mipmap creation at a run time, thus speeding up the original algorithm. This process can be done in the same rendering pass using the capabilities of hardware accelerated vertex shaders in OpenGL, which allows access to different levels of the multiresolution pyramid simultaneously.

4.2 Local Saturation, Contrast and Brightness Algorithms

In a similar manner, we handle local saturation, contrast and brightness. For each operation, the system passes the visual field map to the amount of brightness for each pixel or the luminance for each color channel respectively (this allows for modeling of spatial variations in color vision). The system passes the POR coordinates to the shader in order to achieve proper alignment between the FBO texture and visual field map. An example of such a shader is provided in Listing 1. Each image processing operation can be performed separately or combined as shown in Figure 2.

```
uniform sampler2D OriginalTexture;
uniform sampler2D ResulutionMap;
varying vec2 texCoord;
void main(void)
{
   float grayAmount = texture2D(ResulutionMap, texCoord).a;
  vec3 lumCoeff = texture2D(ResulutionMap, texCoord).rgb;
  vec3 textureColor = texture2D(OriginalTexture,texCoord).rgb;
  vec3 intensity = vec3 (dot(textureColor.rgb, lumCoeff));
  vec3 color   = mix(intensity, textureColor, grayAmount);
  gl_FragColor = vec4 (color, 1.0);
}
```

Listing1 : Saturation Shader

4.3 Spatial Distortion Algorithm

The algorithm for spatial mapping is based on bump mapping shading techniques [Rost, 2004]. Usually this technique is used to simulate texture details on a flat surface. By doing this, the system can achieve a very high level of detail at a low CPU power since all the calculations are done in a single pass to the fragment shader. Our method deforms the surface at the pixel level by using a normal map that describes elevation and distortion of the surface. This normal map can be coded into color channels and hence our system uses a visual field map that describes the visual 'indentations' that a visually impaired patient can experience.

4.4 Glare Algorithm

The algorithm uses high dynamic range rendering techniques (HDR) to generate glare and glow of objects [Ward 1994]. Achievement of this algorithm requires several graphical passes. On the first pass, the system creates a smaller version of virtual image. On the next pass, the smaller version is blurred. On the final pass, the smaller image is convolved back with the original image. During the convolution the system uses the visual field map to control the amount of exposure in each area.

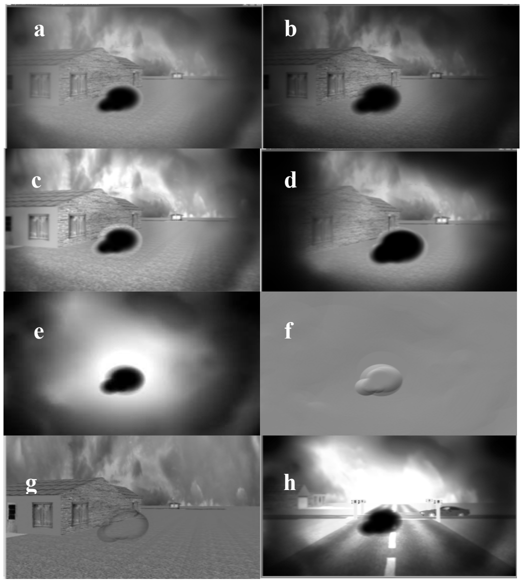

Figure 2: a. Contrast; b. Brightness; c. Saturation d. Combined Image; e. Resolution Map; f. Normal map; g. Distortion; h. Glare

5 System Set-up

The head-eye tracking system is based on a video eye tracking subsystem (VISION2000, EL-MAR Inc., Toronto ON) and an acoustic-inertial head tracking subsystem (IS-900, InterSense Inc., Burlington VT). The head tracker is rigidly mounted to the eye tracker head gear. Both the head tracker and eye tracker work at a synchronized frequency of 120Hz and the data is sent to the workstation over RS232 serial ports at a baud-rate of 38400 and 19200 respectively.

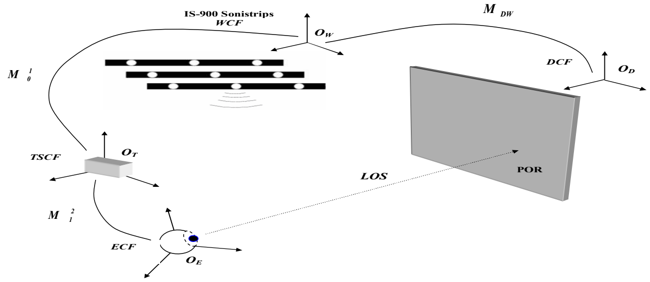

Figure 3: Coordinate frames of the combined system and their relationship; ***WCF*** *- World coordinate frame;* ***TSCF -*** *Tracker station coordinate frame;* ***ECF -*** *Eye coordinate frame;* ***DCF -*** *Display coordinate frame*

There are four coordinate frames used in this combined system: the IS-900 transmitter frame which is consistent with the world coordinate frame; the IS-900 tracker station frame; the eye coordinate frame and the display coordinate frame. The notations of each frame and their origins are shown in Figure 3.

As the system supports free head and body movement it can be used in a variety of settings and with different kinds of displays. One of the simplest settings can be when the subject is working with images shown on a monitor. For example, such a configuration could be used with a monitor for reading tasks or face recognition studies in age-related macular degeneration patients. Experiments that require a larger visual field such as driving simulations can be conducted in front of a video wall display. In addition, the tracking system is flexible enough that it can be used in an interactive CAVE-like environment such as the Immersive Virtual Environment at York (IVY) [Robinson et al. 2001]. IVY is a fully immersive (six-sided) projective stereoscopic visual environment that can be used for a range of tasks from structure visualization to study of human perception in real and virtual environments.

In order to support different types of displays, a custom developed API, the Virtual Environment library (VE), has been used to abstract the various display and input technologies required by applications. VE provides a framework for screen geometry and positioning with parallel rendering of screens on multiprocessor machines, a general input device mechanism, an event-based programming model and a simple language for specifying the physical setup of the environment, including screens, input devices and user-specific parameters [Robinson at al. 2001].

To maximize flexibility the system uses script files to configure the virtual environment at run time.

Experiments were run under the Linux operating system on a PC with a 3.2 GHz Pentium 4 processor and 4 GB of RAM. With this system and an Nvidia GeForce 8800 to support our image processing we can easily achieve 16 ms processing latency (and 60 Hz frame rate) even with multiple numbers of passes.

Conclusions

A gaze-contingent projective virtual environment was described that enables the study of the effects of low vision on everyday tasks such as driving and navigation. It also provides an opportunity to study behavioral patterns in simulated early stages of glaucoma and macular degeneration. The real time simulation system relies on a calibrated combined head–eye tracking system that can calculate the line of sight as well as the point of regard on different kinds of displays while allowing for free head and body movements. Future work will focus on further development and evaluation in controlled user studies and application of the system to study the effects of visual field defects on driving and navigation.

References

ALLISON, R.S., M. EIZENMAN, AND B.S.K. CHEUNG. 1996. Combined head and eye tracking system for dynamic testing of the vestibular system. *IEEE Transactions on Biomedical Engineering*, 43(11), 1073-82.

ALLISON, R.S., EIZENMAN, M., TOMLINSON, R.D., NEDZELSKI, J. AND SHARPE, J.A. 1997. Vestibulo-ocular reflex deficits to rapid head turns following intratympanic gentamicin instillation. *Journal of Vestibular Research*, 7(5), 369-380.

BRIDGEMAN B., HENDRY D., AND STARK, L. 1975. Failure to detect displacement of the visual world during saccadic eye movements. *Vision Research*, 15: 719-722.

DODGE, R. 1900. *Visual perception during eye movement. Psychological Review*,7, 454-465.

DUCHOWSKI, A. T. 2002. A Breadth-First Survey of Eye Tracking Applications. *Behavior Research Methods, Instruments, and Computers ,* BRMIC, 33 (4), 455-470.

DUCHOWSKI, A. T. 2003. Gaze Tracking Methodology: *Theory and Practice.* London, UK: Springer-Verlag (London), Inc.

DUCHOWSKI, A. T., AND COLTEKIN, A. 2007. Foveated Gaze-Contingent Displays for Peripheral LOD Management, 3D Visualization, & Stereo Imaging. *ACM Trans. on Multimedia Computing, Communications, and Applications,* 3, 1-21.

FORTENBAUGH ,F.C. , HICKS, J. C., HAO, L., AND TURANO., K. A. 2007. A technique for simulating visual field losses in virtual environments to study human navigation. *Behavior Research Methods*, 39 (3), 552-560.

GEISLER, S.W., AND PERRY, J.S. 1998. A Real Time Foveated Multiresolution. *System for Low-Bandwidth Video Communication Display.* Proc. SPIE, 3299, 294–305.

GEISLER, W.S., AND PERRY, J.S. 2002. Real-time Simulation of Arbitrary Visual Fields. *Proc. ETRA 2002*, ACM Press (2002), 83-87.

HUANG, H. A. Calibrated Combined Head-Eye Tracking System. *York* , Toronto. 2004.

LEIGH, R.J., AND ZEE, D. S. 1991. *The neurology of eye movements. Philadelphia: E A. Davis*.

NIKOLOV, S.G., NEWMAN, T.D., BULL, D.R., CANAGARAJAH, N.C., AND JONES, M.G. 2004. Gaze-Contingent Display Using texture mapping and OpenGL: System and application. *Proceedings of the 2004 symposium on Eye tracking research and applications*, 11-18.

Perry, J.S., and Geisler, W. S. 2002 .Gaze-contingent real-time simulation of arbitrary visual fields. *Human Vision and Electronic Imaging , SPIE*, 4662, 57—69.

ROBINSON M, LAURENCE J, ZACHER J, HOGUE A, ALLISON R, HARRIS LR, JENKIN M, AND STUERZLINGER W. *2001.* Growing IVY: Building the Immersive Visual environment at York. *ICAT. Proc. 11th Int. Conf. on Artificial Reality and Telexistance, Deformation, The Visual Computer 16, 1, 47-61.*

ROBINSON R. M., THOMAS M. L., AND WETZEL P. A. 1989. *Eye Tracker Development On The Fiber Optic Helmet Mounted Display*. Tech. rep., Air Force Human Resources Laboratory, Operations, Training and University of Dayton.

ROST, R. J. 2004. *OpenGL Shading Language*. Addison-Wesley.

WARD, G. J. 1994. The RADIANCE lighting simulation and rendering system. *In Proceedings of SIGGRAPH '94, A. Glassner, Ed.*, 459–472.