Musings on Volumetric Level of Detail for Virtual Environments[†]

Martin Reddy

Dept. of Computer Science
University of Edinburgh
Mayfield Road, EH9 3JZ

e-mail: `M.Reddy@ed.ac.uk`

Abstract: This paper considers the extension of a real-time graphics renderer to support fovea enhancement—the technique of localising visual detail to the particular region of the display which the user is looking towards. This is performed by extending the standard notion of distance Level of Detail (LOD) into three dimensions to give volumetric LOD; whereby the LOD of an object is related to its presence within a three- dimensional volume which is aligned with the user's gaze. Before introducing this technique, some background details are discussed regarding the relevant characteristics of the human visual system and current solutions for effective gaze tracking. Subsequently, a brief cross-section of relevant research is presented and a conceptual model of volumetric LOD is formulated. Finally, implementation factors for such a system are considered and a theoretical evaluation is proffered.

Keywords: *eye-tracking, fovea enhancement, level of detail, polygonal complexity, visual acuity.*

1 Introduction

Most current virtual reality (VR) graphics engines are polygon-based, i.e. all objects are constructed from groupings of simple planar surfaces, very often triangles[1]. If an object contains more polygons, then it can be made to appear more detailed and hence more realistic. However, the CPU-time required to process and display an object is proportional to the number of polygons contained within it, i.e. we have a detail/performance trade off. Therefore it is necessary to find the optimal balance of these two antagonistic factors: we want to produce realistic models, but they must also be presented at suitably interactive frame rates.

One common technique which is used to modulate this detail/performance threshold in real-time is the notion of distance Level of Detail (LOD). The basic premise behind this approach is that when a small object is rendered on any computer screen, it can only be displayed to a certain degree of accuracy and detail (due to the limited resolution of the display device). Therefore, it makes no sense to display a highly-detailed object under these conditions because all of the sub-pixel detail will simply not be visible. In order to take advantage of this effect, a number of representations of the same object are created, varying in polygonal complexity (see Figure 1). Then we can use a high LOD model (many polygons) when an object is close to the observer, but substitute this with a cruder model (less polygons) as it progresses away from the observer; thus we gain a computational advantage when processing

---

[†] This article was first published in *Virtual Reality: Research, Development and Application* (1995) Vol. 1, No. 1, pp. 49–56; a publication of the Virtual Reality Society. (c) Virtual Press, 147a High Street, Waltham Cross, Herts, EN8 7LN.

[1] Although parametric surfaces are possible, these are rarely found in real-time graphics renderers due to the much higher computational cost (Deering, 1994).

distant objects without detrimentally affecting the fidelity of the display. This technique has been widely employed in the field of real-time graphics, particularly for car and flight simulators (Economy *et al.*, 1990; Kemeny, 1993; Vince, 1993).

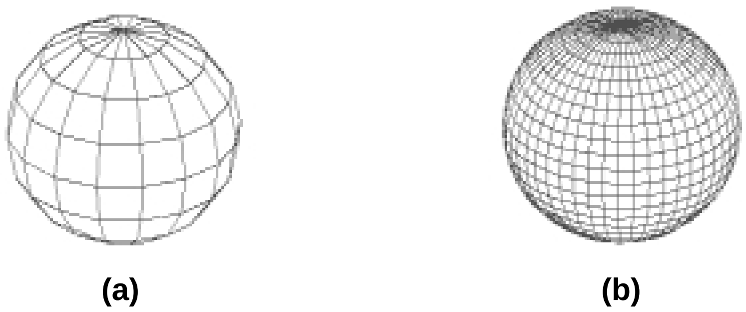

Figure 1: *A simple example illustrating two levels of detail of a polygon sphere. Figure 1(a) contains 112 polygons, and Figure 1(b) contains 1104 polygons.*

The purpose of this paper is to investigate the possibility of extending this technique into three dimensions in order to take advantage of various characteristics of the human visual system. As this is the case, it would seem appropriate to discuss these characteristics in more detail.

2 The Human Visual System

The human eye is a complex and powerful organ which can resolve a sinusoidally-varying contrast pattern with a spacing of up to 0.5 min of arc. However, due to the physiology of the eye, this visual acuity is not uniform across the entire field of view (FOV). This can be explained by analysing the distribution and efficacy of photoreceptor cells in the eye.

There are two types of photoreceptor cells in the retina. These are the rods (which provide high sensitivity in dim light) and the cones (which offer high visual acuity in bright light). In order to gain the best sensitivity/acuity balance, the eye is divided into two regions: a small area rich in cones (the *macula lutea*) and a larger surrounding area rich in rods. The macula lutea contains the *fovea* which is a circular, pit-shaped depression in the retina. The centre of the fovea (the *foveola*) offers a densely packed region of cones; with the cone density decreasing in relation to retinal eccentricity. Beyond 10° from the centre of the fovea, the retina is composed predominantly of rods; providing a wide peripheral vision but with substantially reduced visual acuity (Bruce & Green, 1991).

In addition to this, visual acuity is affected not only by the distribution of rods and cones across the retina, but also by the degree of pooling of these cells. The outputs from the photoreceptor cells are filtered through a layer of several nerve cells (the *horizontal*, *bipolar*, *amacrine* and *ganglion cells*) before being transported via the optic nerve to the brain. A number of rods or cones can be pooled together by this synaptic network to form a single output via the axons of the ganglion cells. If we investigate this relationship, we find that the number of photoreceptors which stimulate a single ganglion cell increases towards the periphery of the retina.

The result of this is that, in general, our visual system has a FOV of around 180° of arc (with binocular overlap), but only a very small part of this FOV can be resolved in the sharpest detail— around 4° of arc (Levoy & Whitaker, 1990)—with a smooth reduction in visual acuity towards the periphery.

Also of interest here are the characteristic movements of the eye. There are three basic types of eye movement: *saccadic*, *pursuit* and *vergence*. A saccade is a rapid movement of the eye (up to 800° per sec) which is made in order to fixate a target onto the fovea. Having achieved this, the smoother pursuit eye movements serve to keep the target in foveal vision as it moves across the retina, while the vergence movements keep the target fixated at both eyes' foveae if there is a change in target distance (Carpenter, 1991). From experimental results, a user's gaze is normally directed towards any objects on a computer screen; often to the centre of a group of adjacent objects, and particularly towards moving objects, or objects which have just entered the visual field (Stampe *et al.*, 1993).

The implication of the above phenomena for computer graphics is that, if we can track the gaze of the user, then we can degrade the detail of the environment which exists in their peripheral vision (and thus gain a performance improvement), without unduly affecting the user's global perception of the scene. This technique is called *fovea enhancement*.

For the purpose of this paper, I shall introduce the term *foveal volume* to signify the region of 3D space which is being focussed onto the user's fovea; and *foveal spot* as the circular, high resolution area which is rendered on the display device.

3 Tracking Technologies

3.1 Eye-Tracking

Obviously any system which attempts to take full advantage of these phenomena must include some kind of eye-tracking technology in order to monitor the user's oculomotor activity. A thorough discussion of all eye-tracking technologies is beyond the scope of this paper; the interested reader is therefore referred to Young and Sheena (1975) for a general study of the area, and Stampe *et al.* (1993) for a more specific discussion of the issues relevant to computer interfaces such as VR.

Modern devices generally offer angular resolutions of less than 1° of arc with possible lags in the order of milliseconds. However, many eye-tracking technologies are too restrictive to be practically incoporated into a VR system; imposing various constraints such as strict lighting conditions, restrained head movements or implanted sensors. Nevertheless, some of the more appropriate techniques include the following:

- Limbus tracking — uses phototransistors and infra-red LEDs mounted on an eyeglass frame to monitor the boundary between the iris and the sclera. These sensors are small and cheap, but are prone to noise.
- Image tracking — involves training a camera on the eye and using a real-time video processor to determine the position of the pupil (the camera is normally mounted on the head to maximise performance). These systems are fairly accurate but tend to be rather expensive and cumbersome.

- Electro-oculography (EOG) — uses electrodes placed beside the eyes to measure the standing potential between the cornea and the retina. Although these systems offer large ranges (around 170°), they are susceptible to noise and drift, and are of questionable worth for accurate gaze-tracking.

3.2 Position-Tracking

In addition to an appropriate eye-tracking technology, some form of position and orientation tracker is normally required in order to resolve the combination of head and eye movements. Such trackers are already used in immersive VR systems and can be implemented using a number of technologies such as mechanical, acoustic, optical and electromagnetic (Hand, 1993). Each of these is described in more detail below:

- Mechanical devices, such as the Boom, offer high resolutions and induce minimal delays (often zero lag) but they restrict the user to a small working volume and are generally encumbering.
- Acoustic trackers use ultrasonic pulses to measure distance (and thus require multiple sensors to resolve position and orientation). They are cheap and readily available but can be susceptible to spurious noise and are line-of-sight devices (i.e. tracking is interrupted if an object passes between the transmitter and the receiver).
- Optical trackers use devices such as lasers, infra-red LEDs, video cameras and photodiodes to measure distance (and so again multiple sensors are required). These systems allow very large working volumes and offer small lags but they have the drawback of requiring line-of-sight operation and may also require controlled lighting.
- Electromagnetic trackers are the most common choice for immersive VR systems, primarily because they are not restricted to line-of-sight operation and also because the sensors are fairly unobtrusive. These trackers generally offer angular resolutions of around 0.05–0.1 deg of arc and can induce delays commonly in the range of 4ms–150ms, depending on the particular system and the degree of data filtering employed.

4 Previous Research

The concept of a fovea enhanced graphics display is not new. A number of researchers have considered this possibility over recent years.

Levoy and Whitaker (1990) constructed a volume rendering application which followed the user's gaze and smoothly varied the resolution of the display based upon the spatial acuity of the retina. The resolution was varied by modulating the number of rays per unit area and the number of samples drawn per unit length of each ray. The tracking system consisted of a NAC Eye Mark (an infra-red LED system accurate to within 3° of arc) and a Polhemus 3SPACE position tracker. The foveal spot which they used subtended 12° of arc, with a surrounding transition region of a further 8° of arc. The results which were obtained on a $256 \times 256 \times 109$ voxel dataset were substantial (a five fold increase in rendering time was experienced).

Sogitec Electronique constructed a flight simulator offering fovea enhanced rendering in a dome-housed projection system (Hurault, 1993). This was achieved by using two projection channels: a large FOV background channel for peripheral imagery and a central high resolution channel for the Area of Interest (AOI). The high resolution inset was then blended into the background channel at the point where the user was gazing. Originally the system used an electromagnetic tracking system to track only the head position (with the high resolution channel occupying 44° of horizontal arc by 38° of vertical arc). However, later an infra-red LED eye-tracking system was added to the system (with an additional head-mounted load of 250g). Hurault claims that this notably enhanced the realism of the simulation and that no latencies where discernible between the movement of the user's eye and the displacement of the AOI.

Funkhouser and Séquin (1993) incorporated a provision for fovea enhancement into their architectural walk-through of Soda Hall. However, they did not have access to a suitable eye-tracking technology, so instead they made the simple assumption that the user was always looking at the centre of the screen. The detail of each object was subsequently based upon its distance from the centre of the screen. No information was provided regarding the exact relationship between object detail and eccentricity.

It can be argued that a fovea enhancement system is more applicable to ray tracing or volume rendering applications because detail can be varied in these systems on a per pixel basis; whereas in a real-time renderer, the atomic primitive is the polygon, with detail being commonly varied at the object level. However, these more accurate methods are also rather more compute-intensive and tend to require fast, expensive machines to operate at real-time speeds (Levoy and Whitaker used a Stellar GS-1000 graphics supercomputer and later upgraded to the Pixel-Planes 5). Therefore, there is merit in analysing the possibility of incorporating fovea enhancement into a polygon renderer because these systems can be implemented on accessible, low-end architectures; but also because they tend to be the most common choice for VR applications.

5 Volumetric Level of Detail

Volumetric LOD is the logical extension of distance LOD in which, instead of modulating LOD by a one-dimensional (1D) distance, an object's LOD is modulated depending upon it's presence within a three-dimensional (3D) volume of space (see Figure 2), i.e. if an object exists inside the volume then it is rendered in high detail; if it exists outwith the volume then it is rendered in low detail. If this volume is shaped to resemble the foveal volume of the eye and oriented with the user's gaze, then we gain support for a fovea enhanced display.

As described in section 2, the fovea is roughly circular. This implies that the shape of a volumetric LOD threshold should be conical (with a slope corresponding to the slope of the perspective projection viewing pyramid[2]). Such a volumetric LOD will implicitly incorporate the standard distance LOD if the base of the conical volume is made to be co-planar with the distance threshold, as illustrated in Figure 2(b).

The method used by Funkhouser and Séquin (whereby the detail of an object is related to its distance from the centre of the foveal spot) is an efficient solution, but it does tend to

---

[2] perspective projection is the rendering technique most commonly employed to transform the 3D environment onto the 2D display device, with a reasonable illusion of depth.

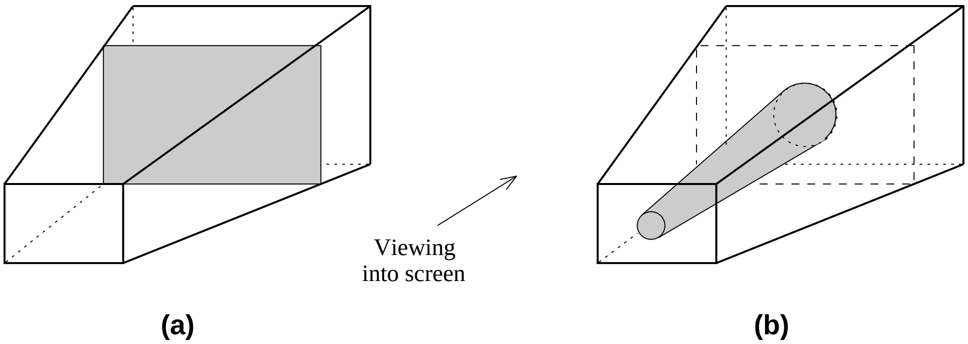

Figure 2: *Comparison of modulation thresholds for (a) distance LOD and (b) volumetric LOD*

simplify the problem. This is due to the fact that it does not take into consideration the extent of the object, i.e. an object's position could be outside the foveal volume, but because of its size, the object may still penetrate the region.

Therefore, the LOD selection criterion must be based upon whether *any* part of the object exists within the foveal volume. This could obviously be a potentially complex and time-consuming operation if it were to be evaluated precisely. Instead, a more time-efficient approximation might be to test for the intersection of the object's bounding box with the foveal volume (a bounding box is simply the smallest box which completely encloses an object. Other simple volumes could also be employed such as bounding spheres). This can be performed much faster than a complete test and, although it is an estimation, it will always be an over-estimation (i.e. an object will always be rendered in high detail if it penetrates the volume, although it may occasionally remain in high detail just outside the volume).

There are a number of ways and degrees in which volumetric LOD could be implemented, a few of which are presented in the remainder of this paper.

5.1 An API-Level Implementation

This section briefly discusses how to implement volumetric LOD without modification to the graphics renderer. This is desirable because an interested party may be using a commercial renderer to which the source code is not readily available. Therefore this section is targeted at the API-level (Application Program Interface—the software interface to the underlying renderer).

For each frame, the new position and orientation of the conic volume must be calculated; based upon the output from the eye and position trackers. The important variables here are the radius of the cone where it intersects the viewing plane (i.e. the foveal spot) and the length of the cone from the viewing plane to it's base (i.e. the depth of the volumetric threshold)—the cone's slope will remain constant.

Once this has been instantiated, then for every object in the environment we have to work out if its bounding box intersects the conic volume. In the spirit of optimisation, we can first of all test for the trivial case where the object's position resides within the foveal volume. If this is not the case, we must then consider whether the extent of the object penetrates the volume. This will require a box/cone intersection test. On the result of this test, we can conclude whether or not any part of an object exists within the foveal volume and subsequently select an appropriate LOD.

5.2 A Renderer-Level Implementation

If one has access to the source code for a renderer, then it can be modified to directly support volumetric LOD. Such an implementation will be far more accurate and efficient than at the API-level because we have access to various internal renderer variables and also because the 3D volume test can be effectively refined to a combination of a 2D test and a 1D test.

For a start, we no longer need to worry about conical volumes because now we have access to the projected screen coordinates for all potentially visible objects (which is essentially what the conical volume attempts to encapsulate). If we then also project the coordinates for an object's bounding box into screen coordinates, all we need to do is calculate whether this (2D) projection penetrates the circle formed by the projection of the foveal volume (see Figure 3).

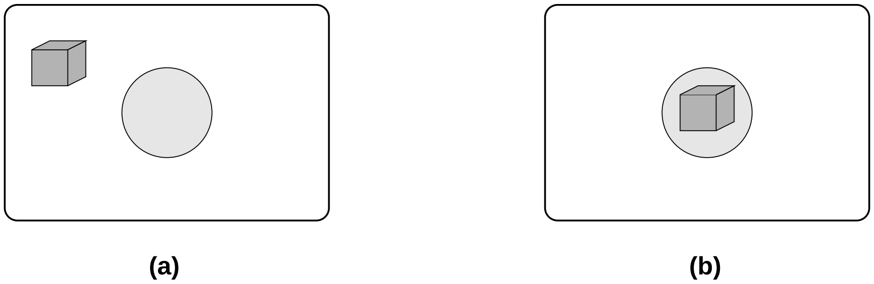

Figure 3: *Illustration of the foveal volume (circle) and a bounding box (cube) projected into 2D screen coordinates. Figure 3(a) shows the case where the object's bounding box is wholly outwith the foveal spot, and Figure 3(b) show the case where it is wholly within the foveal spot.*

Again we can optimise this procedure by first testing to see if the object's projected position lies within the foveal spot. If not then we must consider the extent of the projected bounding box. There is a slight complication to this operation because the projected bounding box will not necessarily be neatly rectangular (it could be hexagonal) and so a more general convex polygon/circle intersection algorithm is required. If this proves inefficient for a particular implementation, then a further optimisation could be employed — a bounding rectangle could be assumed around the polygon so that a simpler rectangle/circle intersection could be calculated first in order to trivially reject those objects which are clearly outwith the foveal spot. Alternatively, a more elegant solution could be achieved if we were to use bounding spheres instead of bounding boxes. In this case, the bounding sphere will be projected into screen coordinates as a circle and therefore a single circle/circle intersection test will suffice to check for penetration of an object into the foveal spot.

The result of this 2D analysis can then be attenuated with a standard distance LOD algorithm to resolve the volumetric component and select the appropriate LOD. This can be performed very efficiently due to the existence of various fast approximations to the 3D Euclidean distance (Ritter, 1990).

5.3 LOD Modulation

Until now we have not considered how large the radius of the foveal spot should be. This will obviously require us to know the horizontal and vertical FOV which the display device occupies[3]. A simplistic approach might then be to select a constant radius related to the size of the fovea (e.g. 12° of arc as used by Levoy and Whitaker).

However, this method takes no account of the perceptual quality of the particular LOD model being selected. We cannot simply select an arbitrarily degraded LOD and expect the user not to notice the switch just because it is displayed outside the foveal volume. This is because the eye's visual acuity does not degrade abruptly beyond the fovea—there is a gradual, linear decrease in visual acuity towards the periphery. Therefore, a more correct system would have to be based upon contemporary models of visual perception.

Our visual acuity can be measured by a quantity called *spatial frequency* which is simply a measure of the sinusoidal variance of contrast over visual angle (specified in units of cycles per degree). As one might expect, the eye's sensitivity to spatial frequency varies across the retina—with high frequencies visible at the fovea but only lower frequencies visible in the peripheral field. Therefore, if it were possible to estimate the spatial frequency content of an object off-line, then based upon the retinal contrast-sensitivity functions (Johnson *et al.*, 1978; Virsu & Rovamo, 1979), we would be able to calculate the correct radius of the foveal spot for each object so that (theoretically) the user would not be able to detect the modulation in LOD.

So far we have simplified matters by only considering the case where we have two LOD models, i.e. we are only required to resolve whether an object is in the foveal region or the periphery. However, multiple LODs are possible. This could be implemented by simply having multiple 'shells' of volumes, one for each LOD.

6 Conclusions

The purpose of this paper has been to propose the technique of volumetric LOD and to discuss the relevant issues involved, such that an interested reader could gain the necessary background to consider a fovea enhanced, real-time, polygon renderer. It should be noted however that no implementation of this technique currently exists. As a result, the content of this paper has been mostly conceptual in nature and further work is required. Nevertheless, there are a number of factors which are immediately evident without implementing a prototype system:

1. Because LOD is performed on a per object basis, if any part of the object lies within the fovea enhanced region, then the entire object must be displayed in high detail. This is obviously rather inefficient, but it is a consequence of the LOD approach chosen. This feature implies that the system will work best for environments which contain many small objects, rather than a small number of large objects.

---

[3] in a head-mounted display (HMD) the FOV information is provided by the manufacturer. However for a standard monitor or projection screen, the FOV must be calculated as a function of the display size and viewing distance, e.g. $FOV_{horiz} = 2\tan^{-1}[(width/2)/distance]$.

2. The technique is not restricted to rigid body elements. Deformable objects can be supported as long as all of the deformed representations are contained within the bounding box which is used to test for foveal volume intersection.

3. As Levoy and Whitaker discovered in their gaze-directed volume renderer, the initial hope of degrading the scene without the user being able to detect any drop in detail, may prove to be a little optimistic. That is to say, under certain circumstances it is likely that the user will be conscious of the degraded peripheral image. This can be attributed to a number of factors including: inaccuracy or drift in the tracking systems, signal filtering latencies, computational lags between successive frames etc.

4. Because the image is optimised upon the eye movements of the user, this means that the system can only support one user per display device. This may restrict the possible uses of a fovea enhanced system; however, valid application areas include: vehicle and flight simulators, visualisation, entertainment systems and most immersive simulations.

5. With regards to immersive VR applications, the FOV of the HMD will affect the performance gain for any particular system such that larger FOV displays will potentially enjoy a greater performance gain (because more of the display will be in the user's peripheral vision). Many of the modern LCD-based HMDs which are currently emerging offer relatively small FOVs (e.g. around 45° of horizontal arc). However when we recall that the eye can only resolve about 4° of arc in full detail, then we would predict that even with these smaller displays we will still experience a suitable performance gain.

Having noted all of the above, it is fair to comment that such a system will offer an appreciable performance advantage for VR applications—assuming that the first point is taken into consideration. Also of particular consequence is the relative simplicity in which volumetric LOD can be implemented. Practically all popular real-time renderers support (or have the ability to support) various LOD models per object. Therefore the only extra code which must be written is that which calculates the object's proximity to the foveal volume(s), and the eye-tracker support. At the current time, it is the eye-tracking technology which is the limiting factor; being relatively expensive, susceptible to inaccuracies and requiring frequent recalibration. However, as the technology matures, these factors will hopefully be diminished.

References

Bruce, Vicki, & Green, Patrick R. 1991. *Visual Perception: Physiology, Psychology and Ecology.* second edn. Lawrence Erlbaum Associates. ISBN 0-863-77146-7.

Carpenter, R. H. S. 1991. *Eye Movements.* The Macmillian Press. ISBN 0-333-47429-5.

Deering, Michael F. 1994. Data Complexity for Virtual Reality: Where do all the Triangles Go? *Pages 357–363 of: Proceedings of the VRST'94 conference.*

Economy, Richard, Ferguson, Robert L., Kelly, William A., & Ramos, Pedro P. 1990 (June). Continous Terrain Level of Detail for Visual Simulation. *Pages 144–151 of: Proceedings of the IMAGE V Conference.*

Funkhouser, Thomas A., & Séquin, Carlo H. 1993. Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments. *Computer Graphics Proceedings, Annual Conference Series*, 247–254.

Hand, Chris. 1993 (2 November). *A Survey Of 3-D Input Devices.* Department of Computing Science, De Montfort University. Available electronically from `http://www.cms.dmu.ac.uk/People/cph/VRbib/Tracking/trackp.html`.

Hurault, Frédéric. 1993 (4–6 May). A Head Slaved Visual System for Flight Simulators. *Pages 37–42 of: Proceedings of the International Training Equipment Conference and Exhibition.*

Johnson, Chris A., Keltner, John L., & Balestrem, Frank. 1978. Effects of Target Size and Eccentricity on Visual Detection and Resolution . *Vision Research*, **18**(9), 1217–1223.

Kemeny, Andras. 1993 (4–6 May). A Cooperative Driving Simulator. *Pages 67–71 of: Proceedings of the International Training Equipment Conference and Exhibition.*

Levoy, Marc, & Whitaker, Ross. 1990. Gaze–Directed Volume Rendering. *ACM SIGGRAPH Special Issue on 1990 Symposium on Interactive 3D Graphics*, **24**(2), 217–223.

Ritter, Jack. 1990. A Fast Approximation to 3D Euclidean Distance. *Pages 432–433 of:* Glassner, Andrew S. (ed), *Graphics Gems*, vol. 1. Academic Press, inc. ISBN 0-12-286165-5.

Stampe, Dave M., Reingold, Eyal M., & Grodski, Julius, J. 1993. Operator Gaze Position Control Interfaces: Investigation of Psychophysical and Operational Parameters. *AGARD CP-541.* (NATO publication).

Vince, John. 1993. Virtual Reality Techniques in Flight Simulation. *In:* Earnshaw, R. A., Gigante, M. A., & Jones, H. (eds), *Virtual Reality Systems.* Academic Press Ltd. ISBN 0-12-227748-1.

Virsu, V., & Rovamo, J. 1979. Visual Resolution, Contrast Sensitivity, and the Cortical Magnification Factor. *Experimental Brain Research*, **37**, 475–494.

Young, & Sheena. 1975. Survey of eye movement recording methods. *Behaviour Research Methods and Instrumentation*, **7**.