"Perceptually Modulated Level of Detail for Virtual Environments"

Martin Reddy

Doctor of Philosophy
University of Edinburgh

1997

To Frances, my Mum.

Abstract

This thesis presents a generic and principled solution for optimising the visual complexity of any arbitrary computer-generated virtual environment (VE). This is performed with the ultimate goal of reducing the inherent latencies of current virtual reality (VR) technology. Effectively, we wish to remove extraneous detail from an environment which the user cannot perceive, and thus modulate the graphical complexity of a VE with little or no perceptual artifacts.

The work proceeds by investigating contemporary models and theories of visual perception and then applying these to the field of real-time computer graphics. Subsequently, a technique is devised to assess the perceptual content of a computer-generated image in terms of *spatial frequency* (c/deg), and a model of *contrast sensitivity* is formulated to describe a user's ability to perceive detail under various conditions in terms of this metric. This allows us to base the level of detail (LOD) of each object in a VE on a measure of the degree of spatial detail which the user can perceive at any instant (taking into consideration the size of an object, its angular velocity, and the degree to which it exists in the peripheral field). Additionally, a generic polygon simplification framework is presented to complement the use of perceptually modulated LOD.

The efficient implementation of this perceptual model is discussed and a prototype system is evaluated through a suite of experiments. These include a number of low-level psychophysical studies (to evaluate the accuracy of the model), a task performance study (to evaluate the effects of the model on the user), and an analysis of system performance gain (to evaluate the effects of the model on the system). The results show that for the test application chosen, the frame rate of the simulation was manifestly improved (by four to five-fold) with no perceivable drop in image fidelity. As a result, users were able to perform the given wayfinding task more proficiently and rapidly.

Finally, conclusions are drawn on the application and utility of perceptually-based optimisations; both in reference to this work, and in the wider context.

Acknowledgements

Where do I start! And having started, where do I stop!

The past few years have been rich with experience; incurring the development of many new friends and acquaintances who have, to varying degrees and in numerous contexts, helped me along the twisting and occasionally disheartening road of research. This is my opportunity to thank all these people for their myriad contributions, suggestions, friendly ears, timely distractions, and full-fat chocolate products when I needed them most.

Firstly, thanks must go to my supervisor, Eric McKenzie, for his support and feedback throughout my period of research. Also in the department, I would like to thank many of my fellow doctoral students for their cherished comradeship, e.g. Marcus Marr, Rob Payne, Isabel Rojas, Thomas Zurek, Thomas Schreiber, and many more. In particular, I owe much to Rycharde Hawkes who had the substantial tasks of proof-reading a draft of this thesis and making sure that I went to the gym regularly (to work off the above chocolate products).

I am also deeply indebted to the members of the Psychology Dept. where I spent much of my time. Specifically, I wish to offer special thanks to Simon Rushton and Julie Harris for their expertise, comments, and encouragement on many of the issues surrounding the visual perception content of my work. Similarly, gratitude is due to Ben Watson of Georgia Tech. for his constructive feedback, and to Alina Moga of Tampere University for catalysing innovation.

Out with the work setting, I would like to offer my fondest regards to Laura Coull for her presence and energy over the last three years. I feel that my time here has been more precious as a result of our friendship. My Saturday evenings would have been much more dull (and my Sunday mornings less hazy) if my old friend Colin Howes had not dutifully dragged me off to the pub once in a while. And finally, to my family, I extend my warmest wishes and thanks; just for being there.

Thanks to you all—it's been a ball.

Declaration

I declare that this doctoral thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text. The following articles were published during my period of research. Certain material and concepts from these publications will necessarily be presented within the body of this work.

1. Reddy, M., Watson, B., Walker, N., and Hodges, L. F. (1997). Managing Level of Detail in Virtual Environments: a Perceptual Framework. *Presence: Teleoperators and Virtual Environments*. (In Press)
2. Watson, B., Walker, N., Hodges, L. F., and Reddy, M. (1997). A User Study Evaluating Level of Detail Degradation in the Periphery of Head-Mounted Displays. *Presence: Teleoperators and Virtual Environments*. (In Press)
3. Reddy, M. (1997). The Development and Evaluation of a Model of Visual Acuity for Computer-Generated Imagery. *Technical Report ECS-CSG-30-97*. Department of Computer Science, University of Edinburgh.
4. Reddy, M. (1996). SCROOGE: Perceptually-Driven Polygon Reduction. *Computer Graphics Forum*, **15**(4):191-203.
5. Reddy, M. (1996). Incorporating Models of Early Vision Processing into Virtual Reality. *The Applied Vision Association (AVA) Annual Meeting*, 1-3 April. Department of Psychology, University of Reading. (Unpublished)
6. Reddy, M. (1996). A Measure for Perceived Detail in Computer-Generated Images. *Technical Report ECS-CSG-19-96*. Department of Computer Science, University of Edinburgh.
7. Reddy, M. (1995). A Survey of Level of Detail Support in Current Virtual Reality Solutions. *Virtual Reality: Research, Development and Application*. **1**(2):95–98.
8. Reddy, M. (1995). A Perceptual Framework for Optimising Visual Detail in Virtual Environments. *Proceedings of the FIVE '95 Conference*. Queen Mary and Westfield college, University of London, UK. pp. 189–201.
9. Reddy, M. (1995). Musings on Volumetric Level of Detail for Virtual Environments. *Virtual Reality: Research, Development and Application*. **1**(1):49–56.
10. Reddy, M. (1994). Reducing Lags in Virtual Reality Systems using Motion Sensitive Level of Detail. *Proceedings of the 2nd UK VR-SIG Conference*. Theale, Reading, UK. pp. 25–31.

(*Martin Reddy*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

'Geometry is not reality. Interactivity is reality.'

(Myron Krueger)

1.1 Motivation

Virtual reality (VR): although the term was only introduced in around 1985, the possibility of being able to immerse and interact within a realistic computer-generated world has fascinated computer graphics researchers for decades. In his seminal paper, 'The Ultimate Display', Ivan Sutherland encapsulated his vision of the future when he speculated:

> '*The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal.*' (Sutherland, 1965)

Clearly we are far from attaining this considerable and provocative goal. Nevertheless, the field of computer graphics has advanced considerably over the past three decades. This has seen a progression from simple, monochrome, wire-frame figures to full-colour, lit, shaded, and textured representations. However, in the perpetual strive towards more efficient photo-realistic displays we are now starting to reach a fundamental limit: that of processing power.

VR is an extremely computational demanding paradigm, requiring the simulation and display of a virtual environment (VE) at interactive frame rates. Even with the use of powerful graphics workstations, a moderately complex VE can involve a vast amount of computation, inducing a noticeable lag into the system. This lag can detrimentally affect the user and may therefore severely compromise the diffusion of VR technology. It is widely considered that this problem of lag is the most pressing which currently faces the VR community (NSF, 1992; van Dam, 1993; Kalawsky, 1993).

This thesis presents the work which has been performed in an attempt to reduce these lags. The method which has been adopted is to trade visual complexity for interactivity. This is not a new concept in itself: the notion of level of detail (LOD) is well founded in the field of computer graphics. However this exposition presents a formal and extensive treatment of LOD which has not been attempted elsewhere to date.

Specifically, we will investigate the efficacy of the human visual system in an attempt to quantify the degree of detail which a user can perceive at any instant. Using this information we can then select the optimal LOD for each object in the VE without the user being aware of any degradation in image quality. In other words, we wish to investigate the fundamental limit of visual detail which a real-time computer graphics system need display; and hence optimise the performance of the system.

In summary, if the ambitions of this research programme could be distilled into a single sentence, then they could be encapsulated as follows:

> **To attempt to reduce the lags inherent in current virtual reality (VR) systems by seamlessly degrading the level of detail (LOD) of objects in the virtual environment (VE) based upon principled models of visual perception.**

The remainder of this first chapter will review the relevant background material to enable a complete understanding of the various issues that this statement involves, and to appreciate the decision processes which ultimately led to this objective.

1.2 Lag in Virtual Reality Systems

Every VR system suffers from an inherent lag to one degree or another. The minimisation of this lag is crucial to the development of effective VR systems, and is the key theme of this thesis. Accordingly, the following sections will detail precisely what we mean by the term lag. Then the effects of lag on the user will be assessed. And finally, a number of solutions will be presented which have been employed to reduce these effects.

1.2.1 A Definition of Lag

Lag is the delay which the user experiences when using a VR system. More precisely, it is the period of time between some change being effected on the VE, and that change being reflected on the display device (Bryson and Fisher, 1990). This can be broken down into two principal types of temporal degradation (Bryson, 1993):

1. **<u>Update Rate</u>** : the rate at which the visual display is refreshed. This is also often referred to as the frame rate of an application. A low update rate implies a high degree of lag.
2. **<u>Transport Delay</u>** : the period of time between a user's input and the effect of that input being represented on the display device, e.g. the lag between a user panning their head and the display being refreshed with the new perspective of the VE.

We shall refer to the cumulative effect of both of these delays as **System Lag**. There are a number of contributing factors which can affect the magnitude of the System Lag. These can be divided, in terms of the application program, into three fundamental components (Atkin, 1993):

1. **<u>Sensor Delay</u>** : the delay incurred by the sensing of the real environment, e.g. the tracking of a user's body part such as their head or hand. This constitutes the delay between the user moving a tracked body part and the computer system receiving notification of that movement.

2. **Processing Delay** : the delay incurred by the simulation of the VE, i.e. all of the calculations which must be performed on each object in order to update its state. Examples of this include: collision detection, implementing object behaviours, incorporating dynamics rules, updating representations of users' body parts, etc.

3. **Rendering Delay** : the delay incurred by the display of the VE. Depending upon the specific algorithm which is employed, this could involve various operations such as: clipping non-visible objects, projecting coordinates, back-face removal, depth sorting, accounting for light sources, applying textures, etc.

Normally, only one of these will prove to be the bottleneck for the entire system. However, which of these this is will depend upon the particular system being used as well as the nature and complexity of the VE being modeled.

For a single-processor model of a VR system, System Lag is simply defined as the linear combination of the above three components, i.e.

$$\textit{System Lag} = \textit{Sensor Delay} + \textit{Processing Delay} + \textit{Rendering Delay}.$$

For a multi-processor model, the relationship between these delays can become more complex and will obviously depend upon the degree and type of parallelism employed. As we are only concerned with the general issues here, we will not complicate the discussion at this stage with definitions of lag in multi-processor systems. Suffice to say however that the same components are relevant to both classes of VR systems; although normally a multi-processor system will incur an additional Synchronisation Delay to coordinate the activity of each processor (Wloka, 1995).

1.2.2 Effects of Lag

The degree of lag which a VR system exhibits can manifestly affect the user in a number of adverse ways. The following list attempts to summarise the potential consequences of high lag systems which have been reported to date:

- **Degraded User Performance** : humans are extremely sensitive to visual lags, and as a result our efficiency diminishes when we are forced to work in a high lag environment. For example, lags of between 30–120 ms have been shown to degrade user performance, depending upon the application (Held and Durlach, 1993). Singhal and Cheriton (1995) state that humans can detect latencies of around 100 ms, and will only tolerate maximum inconsistencies in the order of 200 ms. Gregory (1990) also reports that a lag of around 500 ms can seriously degrade hand–eye coordination tasks such as drawing and writing.

Bryson (1993) investigated the effect of both Update Rate and Transport Delay on the user's ability to perform a particular tracking task. He found that the impact of both of these lags was quantitatively similar, and that the user's performance degraded linearly with respect to lag between 0 and 0.5 seconds.

- **Motion Sickness** : delays in the update of visual information have also been reported to cause effects of motion sickness, the symptoms of which include nausea, pallor, and cold sweating (Money, 1970). This is thought to be due principally to the conflict between the user's visual and vestibular senses (Oman, 1993), often referred to as the Vestibular Ocular Response (VOR). Frank *et al.* (1988) have shown that humans can detect head tracking lags of greater than 5 ms and that delays of more than 30 ms can produce effects of motion sickness. Regan (1995) investigated the side-effects of head-coupled immersive VR applications and found a high incidence of motion sickness (61% of subjects in a 20 minute experiment).

  On a similar note, Uliano *et al.* (1986) report that asynchronous visual delays are also a culprit of motion sickness. This would suggest that fixed frame rate graphics systems should be advocated over variable frame rate ones (Hawkes *et al.*, 1995; Helman, 1994).

- **Degraded Sense of Presence** : the term 'presence' (with respect to VEs) is used to define the degree to which a user feels a sense of being physically present within a computer-generated environment. This concept was first developed by Sheridan (1992) and much research has been done subsequently to investigate the various determinants of presence in VEs (e.g. Slater *et al.*, 1995; Witmer and Singer, 1994; Slater and Usoh, 1993).

  The notion of presence has been proposed as a metric to measure the effectiveness of a VR application, based upon whether the user can interact with it in a natural manner. With respect to this, it has been reported that reduced frame rates (less than 15 Hz) can significantly diminish a user's sense of presence within a VE (Barfield and Hendrix, 1995).

1.2.3 Techniques to Reduce Lag

Based upon the above discussion, it is clearly undesirable to produce VR systems with a high degree of lag. As a result, a number of techniques have been developed to reduce System Lag in VR systems. A number of these are discussed below. After each case, an indication is provided towards the component of System Lag which the technique aims to reduce. This is symbolised with one or more of the labels: SENSOR, PROCESSING, or RENDERING.

It should perhaps be noted at this stage that it is not possible to totally eradicate lag: we must perform some calculations in order to process and display a VE, and these will always involve a finite period of delay. However, we can attempt to minimise the magnitude of the lag in order to avoid the above undesirable side-effects. The exact value of what constitutes an acceptable degree of lag is uncertain, although it will undoubtedly depend upon the particular application in question. However, it is generally accepted that a 15 Hz frame rate (i.e. a lag of ~66 ms) is a minimum requirement for interactive, real-time, three-dimensional (3D) graphics systems (e.g. Barfield and Hendrix, 1995; Falby *et al.*, 1993).

- **Motion Prediction** : motion prediction techniques attempt to overcome the sensor delay incurred by sampling the position and orientation of the user's body part, or parts. The underlying principle is to pre-empt the tracking system by calculating estimated intermediate values and then using these approximations until a new reading is available from the sensor. As Kalawsky (1993) notes, simple extrapolation schemes are generally unsatisfactory, so most successful systems have been based upon Kalman filtering techniques (e.g. Liang *et al.*, 1991). [SENSOR]

- **Update Rates** : the notion of update rates[1] can be used to define the rate at which an object's state is recomputed (Wloka, 1993b). That is, to decide upon the amount of processing time to devote to an object.

[1] N.B. update rates in this context is different from the singular term update rate, which was defined previously on page 3.

The premise of this work is that many objects are either slow-moving, relatively unimportant background objects, or static objects and therefore do not require recomputation at every frame. [PROCESSING]

- **Visibility Culling** : in the demanding domain of VR, it becomes necessary to design graphics systems that restrict computation to only those entities which require consideration. Objects which are not within the current viewing frustum should be immediately discarded from the rendering process. Additionally, it is often desirable to discard any polygons from a scene which are back-facing or obviously not visible. This can alleviate much of the unnecessary burden on the graphics system. [RENDERING]

- **Frameless Rendering** : standard double buffered graphics systems can only display a new frame once the entire image has been rendered. In contrast, frameless rendering is a technique which displays each new pixel as it is calculated. Obviously, if the scene were to be rendered in a systematic left→right, top→bottom fashion then the user would be aware of the progressive update of the screen. Instead, pixels are rendered in a pseudo-random order so that the effect is less visually distracting (Bishop *et al.*, 1994). The applicability of this technique to polygon renderers (as commonly used in VR applications) has been demonstrated (Wloka *et al.*, 1995) and shown to increase the frame rate of the graphics system. [RENDERING]

- **Galilean Antialiasing** : Costella (1993) has proposed a hardware-based motion extrapolation system which operates at a pixel level. Basically, this involves a 'smart' frame buffer which, in addition to holding standard information such as colour and z-buffer values, can also record the velocity and acceleration of each pixel. Using this information, the frame buffer can generate intermediate frames by extrapolating the position of moving pixels. This is performed in order to preserve the update rate of the system while the graphics system is generating the next frame. [RENDERING]

- **Level of Detail** : by storing a number of different representations of objects, each varying in complexity, the VR system can select a particular representation to use for each object based upon a number of possible criteria. Using a less complex level of detail (LOD) model will produce a coarser representation than a more complex one, but it will also induce a smaller lag. [PROCESSING, RENDERING]

- **World Subdivision** : the entire VE is segmented into a number of smaller, self-contained volumes. Only objects contained within those volumes which the user is occupying, or can see at any time, are considered for processing and rendering. This technique is only of use if the VE is amenable to this sort of segmentation. For example, it is particularly applicable to architectural walkthroughs where the boundary of each room can be used as the boundary for the subdivision (Airey *et al.*, 1990): but it would be of little use for large, open spaces. [PROCESSING, RENDERING]

- **Employ Parallelism** : as alluded to in Section 1.2.1, we can attempt to reduce lags by using multi-processor systems and parallelising the major processes. This of course does not actually reduce any of the component lags in a system; but it does reduce the combined effect of these, i.e. System Lag. This approach has been used in a number of successful systems such as Division Ltd.'s PROvision systems (Atkin, 1993), Silicon Graphics Inc.'s IRIS Performer (Rohlf and Helman, 1994), and the freely-available MR-Toolkit package (Shaw *et al.*, 1992).

1.2.3.1 Discussion of Solutions

Each of the above solutions offer various advantages and disadvantages. In the case of the motion prediction techniques, such as Kalman filtering and Galilean antialiasing, because we are dealing with a human-in-the-loop simulation, there is only a finite degree to which data can be extrapolated. For example, Liang *et al.* (1991) report that the reliability of prediction decreases as the length of the prediction increases; and that predictions beyond three steps ahead were not feasible.

Graphics optimisation techniques such as visibility culling, frameless rendering and world subdivision provide a means of reducing the burden on the graphics system. However, there is only a certain extent to which this can be performed for any VE, and so these techniques do not provide us with any flexible means to modulate the workload of the system.

With regards to employing parallelism and designing custom architectures to reduce lag, it is the author's opinion that if VR is to become a widespread and accessible technology, then it must be made available on low end, off-the-shelf architectures. For this to be possible, work is required into the development of algorithmic solutions to combat the various latencies currently experienced in VR systems, as opposed to resorting to custom hardware solutions. This view is held by a number of other researchers (e.g. Billyard, 1993).

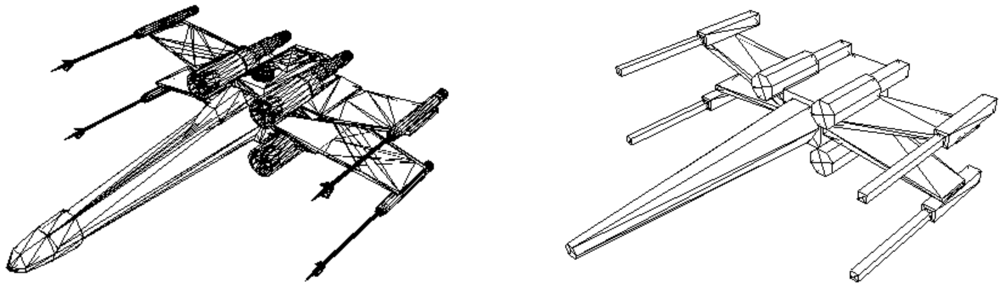
Arguably, the only solutions presented above which enable the application program to balance the load of the system in real-time are: update rates and level of detail. All of the others offer a single (meaningful) reduction in lag for any application. Of these two solutions, we are going to concentrate on the notion of level of detail. The following section will therefore describe this technique in further detail.

1.3 Level of Detail

The complexity of any VE will influence the time required to process and display that environment. For example, a VE which is composed of a large number of polygons will take longer to process and render than a VE with fewer polygons (Deering, 1994). This is not a simple relationship because, for example, the size and geometry of polygons can affect performance, and the use of texture maps can add visual detail to a VE which would otherwise require a large number of small polygons. However, in general, we can state that the more visual complexity that we include in a VE, the greater the lag which will be subsequently induced.

The technique of level of detail takes advantage of this relationship by effectively modulating the visual complexity of each object in real-time. This is done by storing a number of different representations for an object, each varying in complexity, and allowing the graphics system to select the most appropriate model to display at any juncture. Figure 1.1 illustrates this concept with two different levels of detail for a single object.

In essence, LOD techniques attempt to trade spatial fidelity for temporal fidelity: a less complex model will be displayed more rapidly, but it will appear more coarse. This is a valid trade-off for a real-time, interactive graphics system because it has been shown that humans are far more tolerant to reductions in image quality than to delays in visual update (Smets and Overbeeke, 1995).



(a) **(b)**

Figure 1.1: *An example of two different levels of detail for a single object, where (a) contains 6,048 polygons and (b) contains 278 polygons.*

For example, Swartz *et al*. (1992) have shown that frame rate is more important than resolution for target detection, recognition, designation, and tracking.

1.3.1 Generating Level of Detail

There are a number of ways in which we can vary the computational complexity of a polygonal model; and thus create different levels of detail for an object. The most common techniques are as follows:

- **Polygon Reduction** : the goal of this approach is to produce a new model with the same general form and genus as the original model, but containing fewer polygons. This normally implies the invocation of some polygon reduction or simplification process over the original object. A number of these will be discussed in the following chapter.

- **Texture Mapping** : texture mapping techniques can also be used to modulate the geometrical complexity of a model. Essentially, we can replace regions of high geometric detail with a single textured polygon. The polygon's texture is simply a rendered image of that section, from a certain viewpoint and distance. This optimisation can introduce visual artifacts if the model is viewed from a different viewpoint or distance. However, a number of solutions exist to combat this problem; including warping the texture image (McMillan and Bishop, 1995), warping the adjacent geometry (Aliaga, 1996), and smoothly transforming between geometry and texture (Maciel and Shirley, 1995).

- **Illumination Models** : the particular illumination model which is used to render an object can influence the display time. For example, lighting models such as Phong or Gouraud (smooth-shading) produce more realistically shaded results than the Lambertian (flat-shading) model, but at the price of higher computational complexity (Foley *et al.*, 1990). Therefore, we can use different lighting models to render an object to introduce different levels of detail (Funkhouser and Séquin, 1993).

1.3.2 A Typical Implementation

The normal way in which LOD is implemented is to base the model selection upon the distance of the object from the viewpoint. This means that we can use a high LOD model (e.g. many polygons) when an object is close to the observer, but substitute this with a cruder model (e.g. less polygons) as it progresses away from the observer. Thus we gain a computational advantage when processing distant objects without detrimentally affecting the fidelity of the display.

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 (see Figure 1.2).

The term level of detail is often synonymous with distance LOD (because this is the most prolific use of LOD). However, for the purpose of this thesis we should note that LOD is simply a technique to modulate the detail of an object: it does not imply any criteria for selecting an object's detail level, i.e. LOD does not imply distance LOD. As we will see, there are many other criteria which we can utilise to choose the most appropriate LOD at any point.

1.3.3 Identifying and Addressing Problems with LOD

The technique of LOD offers a powerful method to reduce the computational burden of a VR system. However, there are a number of problems with this technique as it stands. These include:

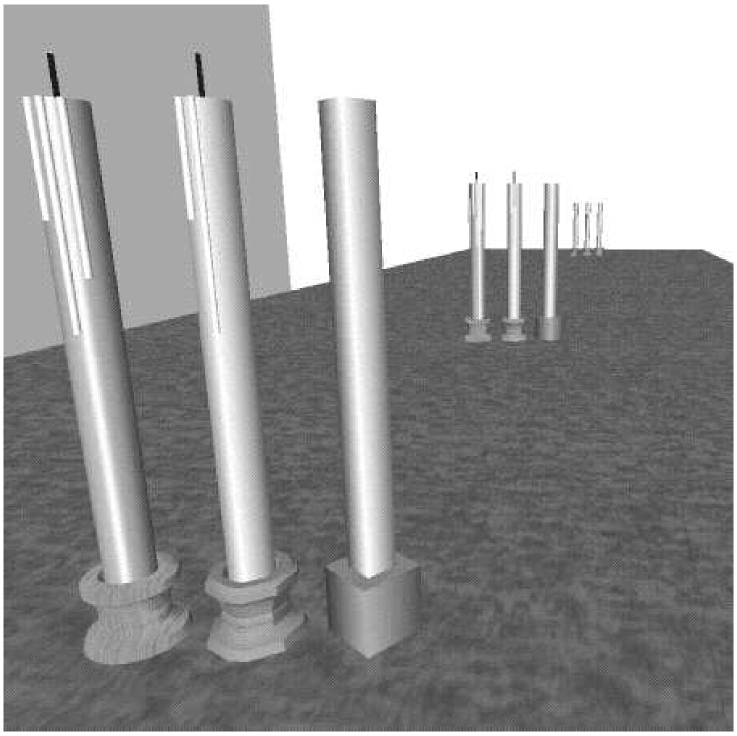


Figure 1.2: *An illustration of distance LOD for three different models of a candlestick. Reproduced with kind permission from Wernecke (1993). (c) Silicon Graphics Inc., 1994.*

1. There is currently no principled mechanism to select the optimal LOD at any point. As a result, most systems use trial and error judgements or *ad hoc* heuristics to decide this (Chrislip and Ehlert Jr., 1995). While this may be satisfactory for the occasional distance LOD implementation, a general LOD paradigm must be based upon a more formal and tractable solution.

2. As a direct consequence of there being no formal procedure for selecting LOD, we find that very often a noticeable flicker is incurred when switching between different levels of detail (the so called **popping effect**). In order to impose the minimum distraction on the user, we desire a system whereby the switch between levels of detail produces no visual artifacts.

3. Various selection criteria have been proposed to manage the modulation of LOD (in addition to distance LOD). These will be discussed later in Section 2.1. However, only a few simplistic attempts have been made to integrate all of these into a single, generic, and orthogonal framework.

4. There is very little support in currently available VR packages to actually generate different levels of detail for an arbitrary object (Reddy, 1995b).

Without the ability to produce various LOD models, the effectiveness of LOD is obviated.

This thesis will attempt to address each of these issues by investigating theories of visual perception. This is done in order to produce a model which quantifies the degree of visual detail that a user can perceive, and then to use this information to select the optimal LOD for each object without the user being aware of any change.

1.4 The Visual System

In order to modulate the LOD of an object based upon its perceptual content, one must first possess an appreciation for the way in which the human visual system is designed and how it is believed to function. The following sections describe the anatomy and physiology of the visual system, providing a basic foundation to help the reader understand and follow the subsequent perceptual material.

The visual system can be divided into three major processing sites (see Figure 1.3). These are the **eyes** (which act as the input to the visual system), the **visual pathways** (which transport and organise the visual information) and the **visual cortex** (the section within the brain which enables visual perception). Each of these sites performs a particular analytical process on the optic image data and will be described in detail in the subsequent sections. Following this, we will develop an appreciation for the spatial resolution of the visual system and how this varies systematically under a number of circumstances.

1.4.1 The Eye

The eye is our window onto the external world, being approximately spherical with a diameter of around 24 mm. The inner **vitreous chamber** is enclosed by three layers of tissue. These are the **sclera**, the **choroid**, and the **retina** (see Figure 1.4).

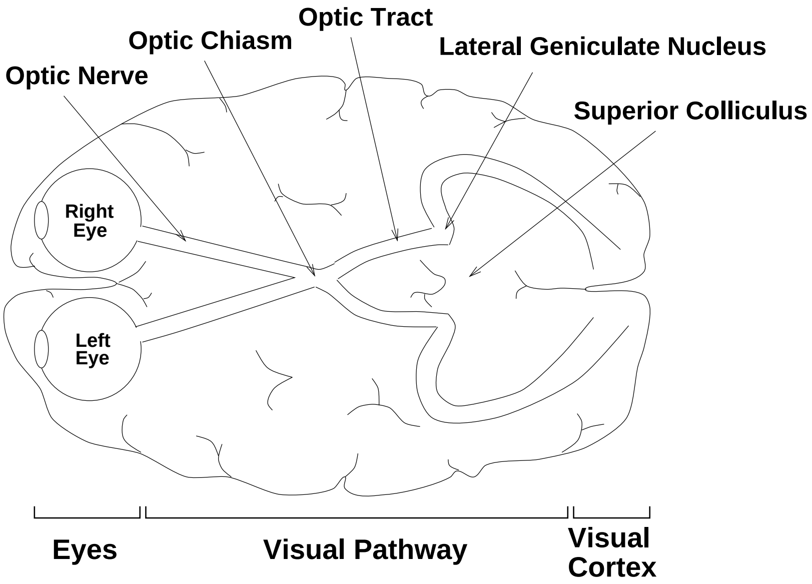


Figure 1.3: *Plan view of the brain showing the extents of the three major sites of processing in the human visual system.*

Of these, the retina is the most pertinent to our discussion because it is the point where light is first detected and processed. As such, its operation and efficiency characterise the raw data which are available to the ensuing vision processes and define the physical constraints and limitations of the human percept system. It would therefore be prudent to investigate the physiology of the retina further.

1.4.1.1 The Retina

Light enters the eye through the transparent cornea. It is focussed by the **lens** and passes through the vitreous chamber before reaching the retina at the back of the eye. Within the retina itself, light must pass through a number of layers of neurons before finally reaching the **photoreceptor cells**. These are responsible for converting the incident light energy into neural signals which are then filtered back through the network of neurons, consisting of the **collector cells** and the **retinal ganglion cells**. The axons of the retinal ganglion cells form the **optic nerve** which transports the neural signals to the visual cortex via the visual pathway. Figure 1.5 illustrates the elements involved in this process.

There are two principal classes of photoreceptor cells in the eye: these are the **rods** and the **cones**. Rods provide high sensitivity in dim light, whereas cones

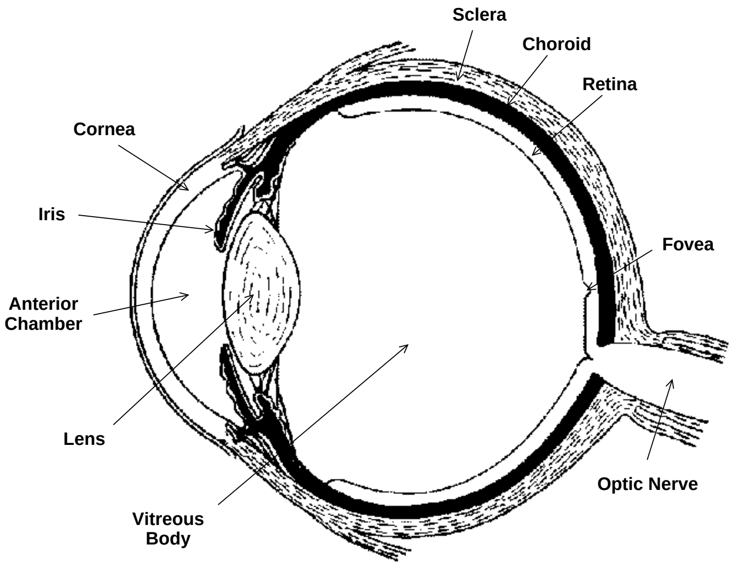


Figure 1.4: *Cross-section of a human eye illustrating its major layers and structures.*

offer high visual acuity in bright light. This duplex arrangement enables humans to see in a wide range of lighting conditions. Both receptor types contain a number of light-sensitive molecules, called **photopigments**. Each of these consist of a large protein called opsin and a vitamin A derivative known as retinal. When light strikes the photopigment, it initiates a reaction which results in the molecule splitting and the subsequent generation of an electric current.

The electrical signals which are generated in the photoreceptors are transmitted synaptically through the collector cells (incorporating the **horizontal**, **bipolar**, and **amacrine** cells) and the retinal ganglion cells. Each eye has roughly 130 million photoreceptors, but only around 1 million retinal ganglion cells. It is therefore apparent that the neural network reduces the retinal image into a more concise and manageable representation—extracting only the relevant parts of the image which are of particular interest. So which features of the retinal image is the eye interested in? In order to answer this question we must examine the visual stimuli which evoke a response from the retinal ganglion cells.

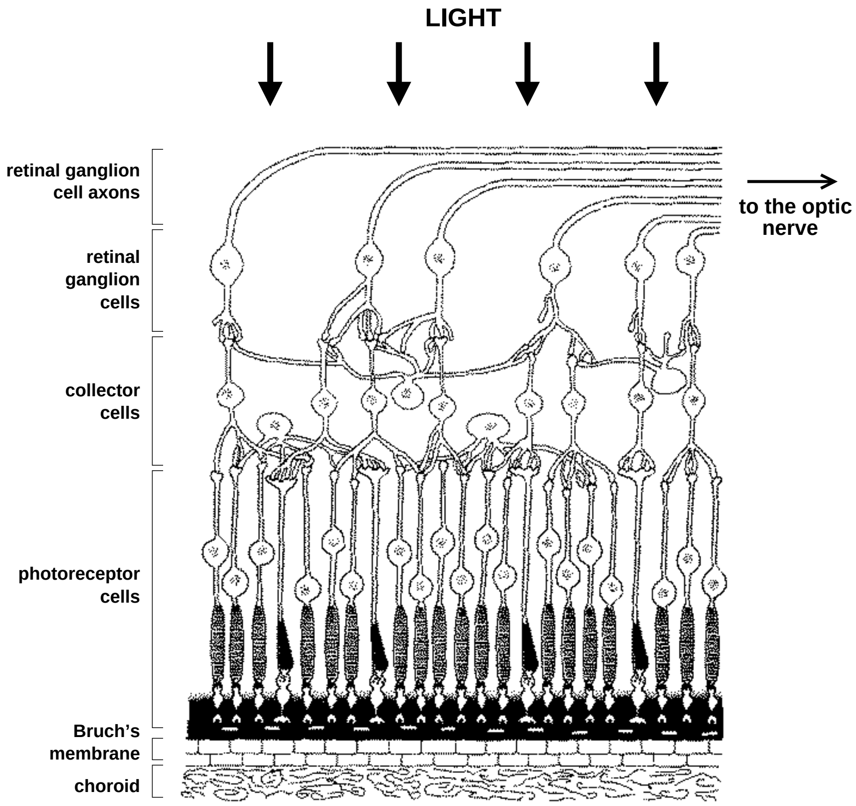


Figure 1.5: *A cross-section of the retina showing the various layers of photoreceptors and neurons which detect and filter incoming light. Adapted from Sekuler and Blake (1994). Note that light must travel through several layers of cells before reaching the photoreceptors.*

1.4.1.2 The Retinal Ganglion Cells

The inputs of the retinal ganglion cells are arranged in an antagonistic, concentric pattern composed of a centre and a surround region. The ganglion cell is continually emitting a background signal; however when light strikes the photoreceptors in one region, this stimulates an increased response from the retinal ganglion cell (a so-called **ON-response**). Whereas light falling on the other region will generate a reduced response, or **OFF-response**. This arrangement is illustrated in Figure 1.6. If the centre region is stimulated by an ON-response, then the retinal ganglion cell is referred to as an **ON-centre** cell. Conversely, if the centre region is stimulated by an OFF-response, then the cell is referred to as an **OFF-centre** cell.

The outputs from the ON-response and OFF-response regions are summed together to form the net response of the retinal ganglion cell. This means that

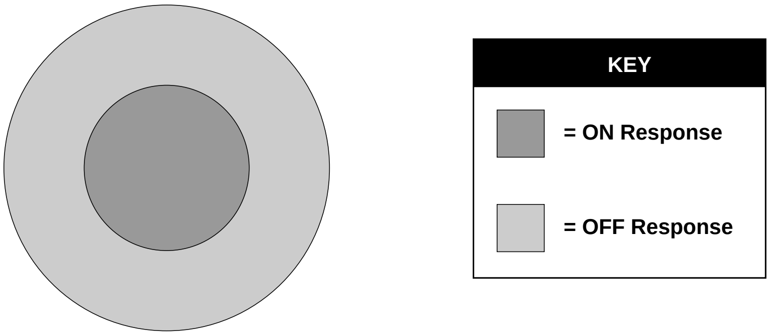
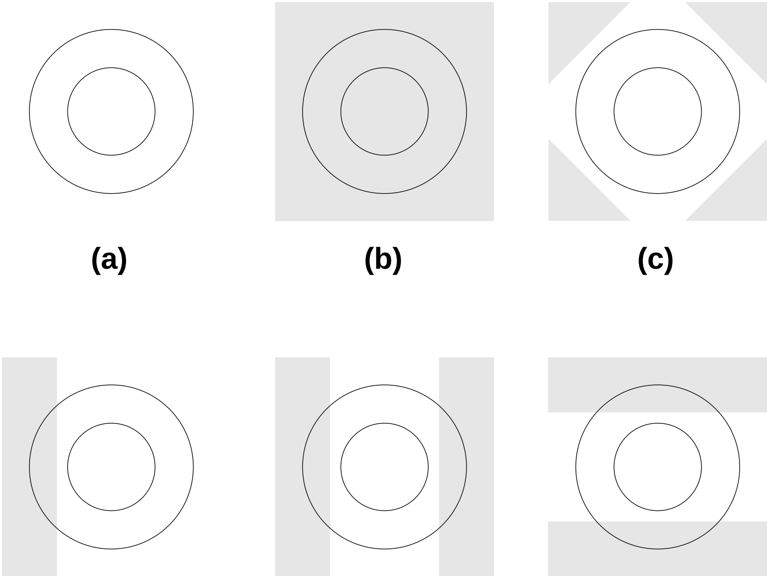


Figure 1.6: *Representation of an ON-centre retinal ganglion cell. Light falling in the ON-response region causes an increase in cell activity; whereas light falling in the OFF-response region causes a decrease in cell activity.*

if the same luminance is presented across the cell then it will elicit a weak response because of the antagonistic reaction between the centre and surround regions. However if an ON-response region receives light when the corresponding OFF-response region does not, this differential will cause a strong response. An illustration of this operation is presented in Figure 1.7.

Accounting for the above observations, we can define the following characteristics of retinal ganglion cells (and hence the first stage of processing which is performed on the retinal image).

- Retinal ganglion cells produce a marked response only when there is a contrast gradient across its **receptive field** (the area of the retina which the ganglion cell receives input from), i.e. they are sensitive to edges.
- A light stimulus which falls out with the cell's receptive field will have no effect on the cell's response.
- The size of the cell's receptive field defines the size of the light stimulus to which it is maximally sensitive.
- The orientation of a stimulus does not affect the cell's response (because the centre and surround regions are circular).



(a) (b) (c)

(d) (e) (f)

Figure 1.7: *Various stimulus arrangements for a single ON-centre retinal ganglion cell. Cells (a)–(c) all generate an equally weak (background) response whereas cells (d)–(f) generate a positive response, with cells (e) and (f) responding equivalently.*

1.4.2 The Visual Pathways

Once the image data leave the eye via the axons of the retinal ganglion cells, they are transported by the visual pathways to the higher vision centre of the brain. However, these pathways do not just passively transport the visual information: some degree of re-organisation and processing is also performed on the signals (although this has relatively little impact on the vision system as we are interested in it).

The main sites of processing on the visual pathways are the **superior colliculus** and the **lateral geniculate nucleus** (LGN). The superior colliculus is believed to be responsible for determining the location of a stimulus and also initiating and controlling eye movements (Carpenter, 1992). Of more interest to our discussion are the lateral geniculate nuclei (there are two—one on each side of the brain), which form a feedback loop with the visual cortex.

The cells of each LGN have antagonistic, concentric receptive fields much like those of the retinal ganglion cells. However, the surround region of an LGN cell exerts a stronger inhibitory response on the centre than its retinal counterpart. This effect induces a more powerful differential between adjacent retinal regions, but contributes very little new to the vision process.

There are two classes of cells in the LGN which remain largely segregated: the **parvocellular** (or P cells) and the **magnocellular** (or M cells). These have a certain correspondence with the cones and rods in the retina, respectively. P cells tend to have smaller receptive fields and so can resolve finer details. They are also sensitive to the colour of light which falls within their receptive fields. By comparison, M cells are sensitive to more coarse features and are stimulated by luminance, not spectral, variances. However, M cells respond vigorously to the rapid motion of a stimulus within the visual field. This provides evidence for the widely-held theory that the various components of vision—form, colour, movement, and depth—are transmitted independently via separate channels to the visual cortex (Livingstone and Hubel, 1988).

1.4.3 The Visual Cortex

The visual cortex (also referred to as the striate cortex, Area 17, and V1) is the major centre of vision. It is located in the occipital lobe, towards the rear of the brain.

As in the retina and LGN, the cells of the visual cortex have a receptive field which restricts the sensitivity of the cell to a certain region. Also, the cortical cells respond maximally to gradients of luminance across their receptive fields rather than ambient illumination levels. However, unlike the retinal and LGN cells, they are also selective on the orientation of a stimulus and the direction of moving stimuli (Blakemore and Campbell, 1969).

We can segregate the cortical cells into two classes (Hubel and Wiesel, 1962): the **simple cells**, which are orientation selective to stationary or slow moving stimuli, and the **complex cells**, which respond maximally to moving stimuli of a particular orientation. As may be envisaged, the receptive fields of cortical cells are not concentrically circular as in the retina and LGN. For example, the receptive field of a simple cell is an elongated shape with discrete excitatory (ON-response) and inhibitory (OFF-response) zones. Figure 1.8 illustrates some examples of how these zones are arranged in order to achieve their orientation selective nature. For example, Figure 1.8(b) will be maximally sensitive to a vertical edge and least sensitive to a horizontal edge. In general, a deviation of around 15° from a cell's preferred orientation is sufficient to render a feature undetectable to that cell (Sekuler and Blake, 1994).

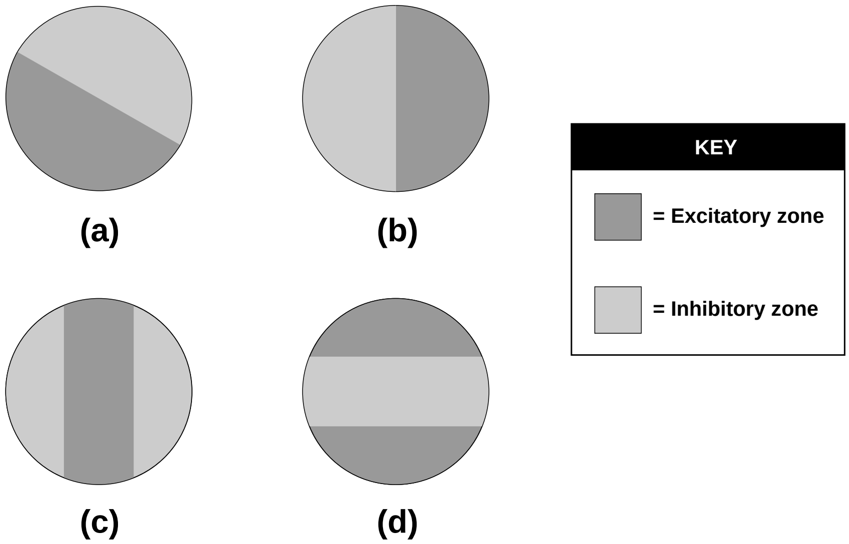


Figure 1.8: *Example receptive field layouts of four simple cortical cells, illustrating the orientation selective nature of these cells.*

Similarly, complex cells are also sensitive to the orientation of a contrast gradient. However, the position of the edge within its receptive field is not as important as it is for simple cells—edges of the preferred orientation can be detected anywhere within the cell's receptive field. In addition, complex cells respond strongly to the presence of a rapidly moving edge. Often this response is selective for a particular direction of movement through the cell's receptive field.

1.4.4 Sensitivity to Visual Detail

We have described the basic physiology of the human visual system. Now let us take a more detailed look at the implications of this design and the effect that these have on the degree of spatial detail that we can perceive.

1.4.4.1 Spatial Resolution

The size of a cell's receptive field (be it a retinal, LGN, or cortical cell) determines the size of stimulus to which it is optimally sensitive. Throughout all three of the vision processing sites we find collections of cells which exhibit a range of receptive field sizes; thus providing sensitivity to a range of stimulus sizes.

The resolution of the human visual system must therefore clearly possess a finite limit which is ultimately determined by the smallest receptive field size. For example, the spacing and pooling of photoreceptors in the retina (which form the inputs to the ganglion cells) will impose the primary limit on how much detail we can perceive. In the most densely packed region of the retina, photoreceptors subtend around 0.5 min of arc. Not surprisingly therefore, we find that the eye can detect detail down to a size of about 0.5 min of arc (Humphreys and Bruce, 1991)[2].

1.4.4.2 Retinal Inhomegenity

The eye's sensitivity to the size of a stimulus is not uniform across the entire retina. Instead we find that a very small part of the retina, known as the **fovea** (see Figure 1.4), has the ability to resolve the smallest features. However this ability degrades in proportion to **retinal eccentricity** (angular distance from the centre of the retina) such that in the peripheral field, the retina has very poor discrimination of fine detail.

This phenomenon means that whenever we wish to focus our attention onto an object, we must perform a combination of head and eye rotations so that the light reflected from that object is projected onto the foveae of our retinae. This ensures that we see the object in the highest detail. Although the peripheral regions of the retina are not as sensitive to visual detail, they are far more sensitive to movement. This provides humans with a highly adapted balance between acuity and sensitivity.

There are a number of physiological features of the visual system that vary with retinal eccentricity. These include:

- The concentration of cells varies dramatically across the retina. In the fovea, only cones are found; and the concentration of these drops off rapidly towards the periphery. Rods appear in large numbers just beyond the fovea, but these also reduce substantially in number towards the periphery. The density of ganglion cells across the retina is roughly proportional to

[2]We will be referring to measures of visual arc quite frequently throughout the text. As these units are often difficult to visualise, it may be illustrative to provide a simple rule of thumb: 1 cm at 57cm distance subtends 1 deg of arc. This roughly means that the width of your thumb in centimeters is its angular subtense in degrees at arm's length. (60 minutes of arc = 1 degree.)

the cone density (Hallett, 1991).

- The receptive field size of retinal ganglion cells increases linearly with eccentricity (Kelly, 1984). This is because the degree of photoreceptor pooling for ganglion cells varies with eccentricity. At the fovea, there is a 1:1 correspondence between cones and ganglion cells. This increases to a 7:1 mapping beyond the fovea (Cowan and Ware, 1985).

- There is a disproportionate number of cells in the visual cortex devoted to the foveal region of the retina. It has been estimated that 80% of all cortical cells are devoted to the central 10 degrees of the visual field (Drasdo, 1977).

The result of these characteristics is that our vision is maximally sensitive within a central region of approximately 5 deg of arc, and drops off smoothly towards the periphery (Zeki, 1993). This reduction in visual acuity across the retina is significant, with around a 35-fold difference existing between the fovea and the periphery (Nakayama, 1990).

1.4.4.3 Temporal Sensitivity

The human vision system cannot resolve as much detail in an object which is moving across the retina as it can in an object which is stabilised on the fovea. This causes the familiar effect of objects blurring as they move past our point of fixation, or as we pan our head to fixate on another target.

The reason for this effect is thought to be due, in part, to the eye's inability to track rapidly moving targets accurately, thus causing a slippage in the retinal image (Murphy, 1978). However, based upon the more recent studies of Tyler (1985), it has been suggested that the photoreceptors themselves are also responsible for limiting our temporal frequency sensitivity (Nakayama, 1990). That is, the process of detecting motion implies an integration of a moving object's stimulus energy over time. Therefore the visual information for precise features in that object are destroyed by this integration process.

1.5 Thesis Content

We have now completed the presentation of required background knowledge. The remaining chapters of this thesis are organised as follows:

Chapter 2 (Background) : the next chapter will present a review of scholarly literature which is relevant to this thesis. This will involve analysing the use and generation of level of detail as well as introducing contemporary models and measures of visual perception.

Chapter 3 (Development) : using the results from Chapter 2, this chapter will develop the base tools and theory which will be required for our model. This includes an appropriate method to generate level of detail, a measure for perceived detail within a computer-generated image, and a model for assessing the degree of detail which a user can perceive.

Chapter 4 (Implementation) : this chapter will describe how the material which has been formulated thus far can be integrated to form a prototype implementation. For example, how the visual metrics can be incorporated efficiently into the graphics system and how each object's level of detail can be calculated from this.

Chapter 5 (Results) : the antepenultimate chapter of the thesis is devoted to a systematic evaluation of the prototype implementation. This involves a suite of empirical studies to assess the merit of a perceptually modulated level of detail system.

Chapter 6 (Discussion) : leading on from the evaluation, we will present an analysis of the results which were obtained and embark upon a discussion of these results; assessing their implications and validity in the wider context.

Chapter 7 (Conclusions) : in the final chapter, we will summarise the result of the research programme, discuss the original contribution to knowledge which has been made, and identify further research activities which have been suggested through the current study.

Due to the cross-disciplinary nature of the thesis content, the initial chapters will necessarily present the background material from the fields of computer graphics and visual psychology as disparate discussions. However as the text develops, this boundary distinction will be eroded until, by Chapter 4, we will see an integrated formulation, where both disciplines merge to form a single, principled solution.

The reader's attention is also drawn to the Glossary of Terms chapter, located immediately after the appendices. This should provide a useful and convenient resource for locating concise definitions of various technical terms used within this dissertation.

The following typographic conventions are adopted throughout the thesis:

Italics is used for figure and table captions, Latin terms, and to emphasise a key word in a sentence.

Slanted Sans Serif is used for any quotations which are replicated verbatim from another source. An attribution will normally be supplied to indicate the source.

Boldface is used in the main text to introduce a new term. Many of these terms can be found in the glossary section towards the end of the thesis.

Boldface Underlined is used for item headings where a list of points is presented.

`Typewriter Font` is used to present any algorithms or sections of C++ source code.

1.6 Summary

Upon reading this exordial chapter, we have gained an awareness of the problems caused by introducing a high degree of lag into a VR system (i.e. reduced user performance, symptoms of motion sickness, and degraded sense of presence). After looking at a number of common techniques to reduce this lag, we have decided to channel our interest towards the notion of level of detail. This is a process whereby a number of different representations for each object are made available to the system, each varying in complexity. The system can then select the most appropriate model to display at any point based upon a particular selection criterion, or criteria.

Following from our examination of the human visual system, we can see that our vision is designed to detect the size and orientation of discontinuities in the retinal image. However, our visual acuity is not uniform under all situations. We have seen that our ability to resolve spatial detail degrades with respect to the retinal eccentricity and angular velocity of a stimulus. This provides us with perceptual justification to base the LOD for an object in a VE upon the degree to which it exists in the user's peripheral field, and the relative velocity with which it is travelling (in addition to the standard approach of distance-based modulation).

Chapter 2

Background

*'Hobbits delighted in such things, if they were accurate:
they liked to have books filled with things that they already knew,
set out fair and square with no contradictions.'*

(J. R. R. Tolkien, *The Lord of the Rings*)

This chapter will present a review of the published literature in three related areas. First we will analyse the field of LOD; looking at the various ways in which LOD has been employed by researchers in the past. Complementary to this is the topic of automatic LOD generation. We will therefore also analyse various techniques to perform polygon reduction on a 3D model. Finally, we will investigate contemporary theories of visual perception and present a metric to quantify the degree of spatial detail that a user can perceive. This information will put us in a position to suggest an efficient system for basing LOD upon principled models of visual perception.

2.1 Level of Detail Implementations

The principle of using multiple geometric representations of objects in order to improve performance was first introduced by Clark (1976). Since that time, LOD has become a commonplace technique in time-critical visual simulations.

Originally LOD was only modulated based upon an object's distance from the viewpoint. However, more recently, other criteria for modulating LOD have been proposed. The following list summarises the five principal criteria currently used to modulate LOD in the field of computer graphics:

1. **Distance** : an object's LOD is based upon its distance from the observer.
2. **Size** : an object's LOD is based upon its pixel size on the display device.
3. **Eccentricity** : an object's LOD is based upon the degree to which it exists in the periphery of the display.
4. **Velocity** : an object's LOD is based upon its velocity relative to the user, e.g. its velocity across the display device.
5. **Fixed Frame Rate** : an object's LOD is modulated in order to achieve and maintain a prescribed update rate for the simulation.

The subsequent sections will take each of these categories in turn and present an overview of the relevant published literature, detailing the various techniques that have been developed to date. This is then followed by a discussion section which will summarise the current state of the art in each area and subsequently highlight the relevant requirements for implementing LOD based upon models of visual perception.

2.1.1 Distance Level of Detail

2.1.1.1 Overview

Distance LOD is when an object's representation is selected based upon the Euclidean distance between the viewpoint and a predefined point inside the object's volume. This is by far the most common and simple usage of LOD. As we have already stated in Section 1.3.2, the theory behind this approach is that as an object progresses further away from the viewpoint, fewer of its high detail components are visible; hence we can select a lower LOD model without greatly affecting the fidelity of the image.

2.1.1.2 Developments

There has been a multitude of applications developed over the past twenty years which employ distance LOD techniques. It would be impractical and fruitless to list them all here, so instead we shall restrict our discussion to notable developments which characterise the genre.

Flight simulators have often used distance LOD techniques to optimise the graphics system throughput (Vince, 1993; Yan, 1985); and these were perhaps the first applications to use the technology heavily. In his extensive survey of the early flight simulator field, Schachter (1981) discusses the need to optimise the number of graphics primitives representing a scene, and states that it is common to display objects in less detail as they appear further away.

In a similar vein, distance LOD has also appeared in vehicle simulators. For example, Kemeny (1993) describes the 'Simulator for Cooperative Automotive NetwoRk' (SCANeR) project: a networked driving simulator developed by Renault, France. This system altered the LOD of objects according to the observation distance and the visibility conditions (i.e. in heavy fog the user can see less, so objects can be degraded sooner). The SCANeR project also incorporated custom LOD generation software which will be described later in Section 2.2.2.6.

Also in the field of simulation, the NPSNET group at the Naval Postgraduate School in Monterey have been working on large scale, distributed VEs for a number of years. Their NPSNET system uses a terrain dataset of 50×50 km at a resolution of 125 m; employing a terrain paging algorithm in order to manage the swapping of visible terrain tiles (Falby *et al.*, 1993). Three different resolutions of this dataset were generated (at resolutions of 250, 500, and 1000 m) and the terrain resolution at any point was based upon a measure of its distance from the viewpoint.

The NPSNET group have also done work regarding the integration of dismounted infantry into their system. Chrislip and Ehlert Jr. (1995) implemented a distance LOD system in order to increase the number of infantry personnel by over seven fold, whilst still maintaining a 10–15 Hz frame rate. They generated four different levels of detail for each figure and also used various environmental conditions (e.g. clouds, fog, smoke, haze, etc.) to slacken the LOD distance thresholds.

2.1.1.3 Associated Techniques: Hysteresis

Astheimer and Pöche (1994) describe the use of hysteresis as a means of reducing the scintillating effect which occurs when an object constantly switches between two different representations at the threshold distance.

Hysteresis is simply a lag which is introduced into the LOD transitions so that objects switch to a lower LOD slightly further away than the threshold distance, and switch to a higher LOD at a slightly closer distance. Figure 2.1 illustrates this concept. In their work, Astheimer and Pöche (1994) found that a hysteresis of 10% of each LOD's range produced favourable results.

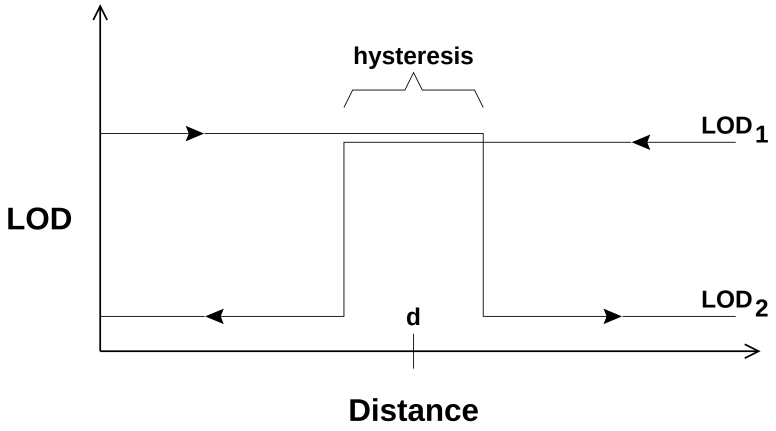


Figure 2.1: *Distance-based switching thresholds between two LODs illustrating a period of hysteresis around the threshold distance, d. Adapted from Astheimer and Pöche (1994).*

2.1.2 Size Level of Detail

2.1.2.1 Overview

Size LOD is when an object's representation is selected based upon the value of its projected size (or area) in screen coordinates. This is therefore simply another way of implementing distance LOD, as objects which move further away will appear smaller on the display device. However, this method offers a number of distinct advantages over the distance-based approach. These benefits will be discussed later in the chapter.

2.1.2.2 Developments

The Open Inventor graphics toolkit from Silicon Graphics, Inc. (SGI) provides a means to automatically select different levels of detail based upon a screen area criterion. The screen area is calculated by projecting the 3D bounding box for a model onto the viewport and then computing the area of the screen-aligned rectangle surrounding that bounding box (Wernecke, 1993).

The REND386 package, originally written by Bernie Roehl and Dave Stampe of the University of Waterloo, provides a real-time 3D graphics library for the IBM–PC compatible platform. The system implements a size LOD system by calculating the projected radius of an object's bounding sphere (Roehl, 1995). This provides a very lightweight and effective mechanism for representing the size of an object in screen space. It also provides an orientation invariant method because the projected radius will be the same length for all object orientations.

Size-based LOD techniques have proved particularly popular for optimising digital terrain models in real-time. These data can often contain vast numbers of vertices and are therefore a prime candidate for LOD reduction. Lindstrom *et al.* (1995) describes one such system in which the terrain model is recursively subdivided into quarters using a quad tree structure; thus, more terrain detail is defined the deeper you descend through the tree. To decide how far down the quad tree to progress for any region of the landscape, Lindstrom *et al.*'s system checks to see if the projected distance between adjacent vertices falls below a predefined pixel threshold. Their system also considers the roughness of the terrain, using a lower polygon resolution where the topography is more smooth. Other similar techniques have also been developed for continuous terrain LOD (e.g. Economy *et al.*, 1990; Scarlatos, 1990).

2.1.3 Eccentricity Level of Detail

2.1.3.1 Overview

Eccentricity LOD is when an object's representation is selected based upon the degree to which it exists in the visual periphery. Without a suitable eye tracking system, it is generally assumed that the user will be looking towards the centre of the display, and so objects are degraded in relation to their displacement from this point.

2.1.3.2 Developments

Funkhouser and Séquin (1993) incorporated a provision for eccentricity LOD into their architectural walkthrough of Soda Hall. 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 display eccentricity.

Hitchner and McGreevy (1993) produced a generalised model of LOD for the NASA Ames Virtual Planetary Exploration (VPE) testbed. They introduced two terms: **detail** (representing the number of geometry primitives per unit area), and **interest** (a measure of the importance of the object to the user). The interest term could incorporate a number of factors, one of which accounted for eccentricity via an equation of the form:

$$interest = \gamma_{\text{static}} / distance, \tag{2.1}$$

where $distance$ is measured in 2D screen coordinates and γ_{static} is an arbitrary scaling factor. Whilst this equation models the general behaviour of the eye's peripheral sensitivity, the units are completely arbitrary.

Similarly, Ohshima *et al.* (1996) developed a head-tracked desktop system that could degrade the LOD of objects in the periphery. They modeled the decline of visual acuity with eccentricity using an exponential relationship as follows:

$$f(\theta) = \begin{cases} 1, & \text{when } 0 \leq \theta \leq \alpha \\ \exp\left(-\frac{\theta-\alpha}{c_1}\right), & \text{when } \alpha < \theta, \end{cases} \tag{2.2}$$

where θ is the angular distance between the centre of the object to the user's gaze fixation, α is the angle from the centre of the object to the edge nearest the user's gaze, and c_1 is an arbitrary scaling factor which Ohshima *et al.* simply report that they instantiated to 6.2 deg.

Watson *et al.* (1995) performed a user study to evaluate the perceptual effect of eccentricity LOD in head-mounted displays (HMDs). A number of subjects were given a simple search task that required the location and identification of a single target object. The degraded peripheral LOD was simulated by using a display division technique where a high detail inset was blended with a coarse background field. The inset was always located at the centre of the display device. For a number of inset sizes and resolutions, Watson *et al*. report that user performance was not significantly affected by the degraded peripheral display and concluded that eccentricity LOD provides a useful optimisation tool.

In addition to the above LOD examples, other techniques have been used in the field of real-time computer graphics to produce displays with increased detail around the area of interest (AOI). For example, Levoy and Whitaker (1990) developed a volume rendering application which followed the user's gaze and smoothly varied the resolution of the display accordingly. Also, Sogitec Electronique constructed a dome-housed projection screen flight simulator offering fovea enhanced rendering by using two projection channels: a large field of view background channel for peripheral imagery and a central eye-tracked high resolution channel for the AOI (Hurault, 1993).

2.1.4 Velocity Level of Detail

2.1.4.1 Overview

Velocity LOD is when an object's representation is selected based upon its velocity relative to the user's gaze. Again, without a suitable eye tracking technology, this must be approximated to the velocity of an object across the display device.

2.1.4.2 Developments

Funkhouser and Séquin (1993) incorporated a velocity term into their architectural walkthrough. They acknowledge that objects moving quickly across the screen appear blurred, or can be seen for only a short period of time, and hence the user may not be able to see them clearly. They therefore reduced the LOD of objects by an amount proportional to the ratio of the object's apparent speed to the size of an average polygon. This latter metric was used as a simple and efficient (although of course perceptually inaccurate) measure of the spatial detail in an object.

Hitchner and McGreevy (1993) also accounted for object velocity in their VPE system so that models with less detail were selected for rapidly moving objects, and models with more detail were chosen for stationary or slow moving objects. This effect was incorporated into their model with an equation of the form:

$$interest = \gamma_{\text{dynamic}}/velocity, \tag{2.3}$$

where $velocity$ is an estimate of visual field velocity, measured by taking the difference between an object's position in consecutive frames, and γ_{dynamic} is an arbitrary scaling factor.

Ohshima *et al.* (1996) modeled the decline of visual acuity with velocity using the following equation:

$$g(\Delta\phi) = \begin{cases} 1 - \frac{\Delta\phi}{c_2}, & \text{when } 0 \leq \Delta\phi \leq c_2 \\ 0, & \text{when } c_2 < \Delta\phi, \end{cases} \tag{2.4}$$

where $\Delta\phi$ represents the angular distance travelled by the object, and c_2 is an arbitrary scaling factor which was simply instantiated to 180 deg/s.

2.1.5 Fixed Frame Rate Level of Detail

2.1.5.1 Overview

Fixed frame rate LOD is when the LOD of each object in a scene is degraded in order to achieve a desired frame rate. This application of LOD is distinct from all those previously discussed because it is concerned with computational optimisation rather than perceptual optimisation.

2.1.5.2 Developments

Any system which implements fixed frame rate LOD must include a scheduler whose job it is to analyse the system load and assign LOD ratings to each object accordingly. There are two main types of scheduler: **reactive** and **predictive**.

A reactive scheduler is the more simple of the two, but it does not guarantee a bounded frame rate. It simply adjusts the detail level based upon whether the previous frame was rendered within the target frame rate, i.e. if the last frame was completed after the deadline, then detail is reduced; and if it was completed before the deadline, then detail can be increased. By contrast, a predictive scheduler estimates the complexity of the frame about to be rendered and enforces LOD assignments to ensure that the update deadline is never exceeded.

The viper system, developed at the University of North Carolina by Holloway (1991), is an example of a reactive system. This attempted to degrade LOD in order to maintain interactive frame rates by simply terminating the rendering process as the system became overloaded. This could cause objects to appear with holes in them, or to disappear completely, as the graphics load increased. One interesting feature of Holloway's (1991) design is the implementation of a simple priority scheme: two priorities (high and low) were used to define whether an object should always be rendered at the highest level of detail, e.g. a representation of the user's hand could be classed as a high priority object and hence should never be degraded.

Funkhouser and Séquin's (1993) architectural walkthrough system employed a predictive fixed frame rate scheduler. They used a cost/benefit paradigm which attempted to optimise the perceptual benefit of a frame against the computational cost of displaying it. That is, given a set S of object tuples (O, L, R) which each describe an instance of an object O, rendered at LOD L and using rendering algorithm R, the overall image quality for each frame was calculated via the following equation:

$$\text{Maximise :} \quad \sum_S \mathit{Benefit}(O, L, R)$$

$$\text{Subject to :} \quad \sum_S \mathit{Cost}(O, L, R) \leq \mathit{TargetFrameTime}. \tag{2.5}$$

The $\mathit{Cost}(O, L, R)$ heuristic was estimated by assuming a two-stage, pipelined rendering model involving a per-primitive processing stage and a per-pixel processing stage. The $\mathit{Benefit}(O, L, R)$ heuristic incorporated a number of factors to determine the object's contribution to model perception, e.g. size, accuracy, importance, focus (eccentricity), motion (velocity), and hysteresis.

Examples of other systems which have used LOD to maintain a desired frame rate include Airey *et al.*'s (1990) architectural walkthrough system [reactive], Hitchner and McGreevy's (1993) VPE testbed [reactive], Wloka's (1993a) Update Rate system [predictive] (see Section 1.2.3), and also the IRIS Performer graphics library developed by Silicon Graphics, Inc. (Rohlf and Helman, 1994) [reactive].

2.1.6 Discussion of LOD Applications

From the above, we can observe that LOD techniques have been used in a vast array of real-time applications. We have seen implementations in flight simulators, vehicle simulators, architectural walkthroughs, visualisations, digital terrain modeling, etc. This confirms that LOD is an extremely useful and general technique which can be applied to most any application in order to evoke a performance advantage.

2.1.6.1 Discussion of Distance / Size LOD

Of the five major criteria used to modulate LOD, distance and size modulation are by far the most prolific. Distance LOD offers two principal advantages. Firstly, it is simple: all that is required is a basic conditional statement to check if a distance exceeds a predefined threshold. Secondly, it is efficient: the only computationally expensive procedure is the calculation of the 3D Euclidean distance between two points; and a number of fast approximations already exist for this (e.g. Ritter, 1990).

However, there are some problems with distance LOD techniques. Principally, we have to chose an arbitrary point within the 3D volume of the object to use for all distance calculations. Therefore, the actual distance between the near side of the object and the viewpoint can change depending upon the orientation of the object and its relative dimensions. This can therefore cause an aggrandisement of any popping effects under certain conditions. Another problem is that if we scale the object to make it larger or smaller, or if we use a different display resolution, then the original distance thresholds are no longer valid and must also be scaled appropriately.

These problems with distance LOD do not occur with size LOD. Size-based techniques provide a measure to determine the visibility of features within an object, regardless of display resolution, object scaling, etc. Also, they obviate the need to select an arbitrary point for the calculation. As a result, size-based LOD techniques provide a more generic and accurate means for modulating LOD than distance-based techniques. The former may be more computationally expensive (because we need to project a number of world coordinates into screen coordinates and then compute the projected size), however Roehl and others have shown that efficient implementations of this are possible.

2.1.6.2 Discussion of Eccentricity LOD

There have been a number of attempts to build computer graphics displays in the past which can reduce spatial detail towards the periphery, both in hardware and software. Within the field of real-time computer graphics and VEs, this has been achieved by modulating LOD based upon an object's displacement from the centre of the display.

To date, all such attempts have employed simple, arbitrary relationships between LOD and eccentricity; such as Hitchner and McGreevy's model in Equation 2.1. These achieve the notional goal of reducing LOD as objects move further towards the periphery; but because they are not founded on any accurate models of visual perception, this degradation is not optimal with regard to user perception, and appreciable popping effects can be experienced as a result. This is reinforced by Watson *et al*. who state that they had no way to predict either the optimal LOD to display or the extent to which LOD could be degraded in the periphery.

In the context of VEs, eccentricity LOD has received interest because of the associated computational benefits which it can offer. However, we should also consider the perceptual costs of such techniques. Watson *et al*. (1995) therefore embarked upon an investigation to assess the perceptual costs of eccentricity LOD. As already stated, they demonstrated that a normal high resolution display offered no significant performance advantage over a display with a low resolution periphery.

We can therefore conclude that eccentricity LOD offers an effective tool to reduce the computational burden of VR systems. However work is still required in order to develop a formal mechanism for selecting the optimal LOD for each object in a scene.

2.1.6.3 Discussion of Velocity LOD

There has been comparatively little work in the area of velocity LOD. The only major contributions to the field are proffered by Funkhouser and Séquin (1993), Hitchner and McGreevy (1993), and more recently Ohshima *et al*. (1996). Each of these employed *ad hoc* heuristics to model the rate of LOD reduction with respect to velocity, with little or no founding in perceptual theory.

In addition, no results are provided to assess the effectiveness of velocity LOD, e.g. Funkhouser and Séquin evaluated their system using an architectural walk-through which contained entirely static components, and so the motion-related part of their algorithm would have played a very minor rôle in the overall system. Also, no consideration has been given to the effect of rotational velocity (e.g. an object which is stationary, but rotating about its axes) and whether this can be used to optimise LOD too.

Certainly, the general concept of using relative velocity to modulate detail on a per-object basis is legitimate. This will automatically support situations such as complex motion flows (e.g. maintaining relative LOD when a moving object crosses a stationary background), and also correctly handle the case when a user is tracking a moving object (i.e. the object is therefore stationary relative to the user and remains in high detail whilst the background is moving and is therefore presented in low detail). However, there is still no formal mechanism to optimally chose the rate at which LOD should be reduced with respect to velocity and, as Funkhouser and Séquin (1993) state, more work is required to resolve this.

2.1.6.4 Discussion of Fixed Frame Rate LOD

Fixed frame rate techniques have evolved significantly over the past few years; particularly with the growth of immersive environments which necessitate a high and consistent update rate in order to maintain the illusion of immersion, and to avoid effects of motion sickness (Wloka, 1993b).

Most currently available systems use a reactive scheduler to provide fixed frame rates because these are simpler to implement. However a reactive system does not guarantee that the threshold frame rate will not be exceeded, e.g. this can become particularly evident during periods where the scene complexity varies drastically from frame to frame. In general, a more sophisticated predictive technique will offer better stress management characteristics.

However, fixed frame rate LOD is not primarily concerned with the perceptual content of the display: it will noticeably degrade the objects in a scene in order to achieve a specified frame rate. This contravenes our initial goal of producing a system which can modulate detail without the user being able to perceive any change. As a result we are not directly concerned with the topic of fixed frame rate LOD in this thesis, and we will not discuss it further here.

2.1.7 A Mandate for Perceptually-Based LOD

We have now looked at the field of LOD and gained an appreciation for current state-of-the-art techniques to modulate detail in a real-time computer graphics system. We will now consider how these techniques could be applied and extended in order to produce a perceptually-based model of detail degradation.

The following points provide a short list of factors which should be addressed in the design of a perceptually-based LOD system:

- **Size or Distance LOD** : for reasons of accuracy, a size LOD approach should be advocated rather than a distance-based system.

  From a purely philosophical perspective, our perception is ultimately based upon the size of stimuli on our retinae, and so we should favour a size-based technique for this reason. However, more pragmatically, we have shown that size-based techniques are more flexible and robust: they involve no arbitrary decisions and are not affected by object scaling or display resolution.

- **Formal Specification of Switching Thresholds** : currently there is no accurate way to select the most appropriate LOD to display at any time. For example, if we take the case of distance LOD: how far away does the object have to progress before we can select a lower LOD, without the user noticing the change?

  In the past, this has normally been done on a trial and error basis (Chrislip and Ehlert Jr., 1995; Maciel and Shirley, 1995; Hitchner and McGreevy, 1993; Funkhouser and Séquin, 1993). However, if we start to introduce switching based upon velocity and eccentricity, as well as size, then a trial and error method is simply not feasible any more. For example, Ohshima *et al*. were unsure whether they should take the product of their velocity and eccentricity scaling factors, or whether a minimum function should be used.

Clearly we need some automatic and principled process whereby the computer can calculate the optimal switching thresholds for each LOD under any visual circumstances.

- **Eye / Head Tracking** : our ability to resolve detail is based upon the velocity and size of objects which are projected onto our retinae. Therefore, to be completely accurate, any perceptually-based LOD system should employ a suitable head and eye tracking technology in order to resolve the user's point of fixation.

  However, it should be noted that Ohshima *et al*. (1996) found that head tracking alone provided very promising results. Also, Watson *et al*. (1995) have suggested that a head-tracked display may be sufficient for many applications. This is particularly encouraging because most eye tracking technologies at the present are unwieldy and suffer from a number of problems such as drift and low resolution. However, this technology will mature with time, and so we should incorporate provision for eye tracking in any perceptual model of LOD.

It is worth pointing out that we should not need to incorporate techniques such as hysteresis or priority schemes into a perceptually-based system. These techniques have been developed to counter the visual anomalies which occur when LOD is used in a non-principled and arbitrary manner. These methods are therefore redundant if we can produce a system which will modulate detail without the user being able to perceive any change.

To date, and to the best of the author's knowledge, no system has yet been implemented which uses contemporary models of visual perception in order to efficiently modulate the LOD of objects in a VE. Despite this, all of the foremost papers in the field note that a perceptually-based system would provide the most accurate LOD mechanism (e.g. Maciel and Shirley, 1995; Funkhouser and Séquin, 1993; Hitchner and McGreevy, 1993). This thesis will therefore illustrate how such a perceptually-based LOD system could be developed.

2.2 Level of Detail Generation

The ability to automatically generate various levels of detail from an original model is obviously integral to the topic of LOD. Without this ability, a user would have to generate these different representations by hand. Such an activity would be a tedious, laborious, and approximate process. However, there are other benefits to be accrued from the simplification of models. These include: reduced storage requirements, more rapid transmission over networks, and the reduction of computational complexity for tasks such as collision detection etc.

The following sections will offer a review of the literature concerning the simplification of 3D geometric models. We shall restrict our discussion to the topic of **polygon simplification** (because practically all VR systems employ polygon renderers as their graphics engine). That is, a process which takes an original polygon description of a 3D object and creates another such description, retaining the general shape and appearance of the original model, but containing fewer polygons. Techniques exist to simplify other geometrical representations, such as spline curves or voxel datasets, however these will not be discussed further here.

Developing a systematic taxonomy for polygon simplification algorithms is not a simple task. This is because there exists a diverse gamut of techniques, and yet many of these incorporate a number of common concepts: hence it is difficult to find a neat or exclusive classification. In the author's opinion, the most successful attempt to provide such a scheme is proffered by Erikson (1996). He suggests three generic categories of polygon simplification: **geometry removal**, **sampling**, and **adaptive subdivision**. We shall define these as follows:

1. **Geometry Removal** : an algorithm that simplifies a model by removing vertices or polygons from its description.

2. **Sampling** : an algorithm that samples a model's geometry and then attempts to generate a simplified model that closely fits the sampled data.

3. **Adaptive Subdivision** : an algorithm that begins with a simple base model and recursively subdivides it, adding further detail to local regions of the model at each iteration.

We will now take each of these three categories in turn and present the various published techniques that fit most favourably into that genus. Each technique will be identified by the title of the article in which it first appeared, along with the authors' surnames.

2.2.1 Geometry Removal

2.2.1.1 Geometric Optimisation (Hinker and Hansen)

Hinker and Hansen (1993) developed a geometry removal algorithm that attempts to merge groups of co-planar polygons into a single polygon. This is done by grouping neighbouring polygons with roughly the same normal. (An adjacent polygon is merged with a group if its normal is within a certain angular tolerance of the average group normal.) Once the grouping has been performed, a new polygon is created from the boundary edges of the grouping, and this is then triangulated to produce the final tessellation.

This approach can produce substantial reductions for over-tessellated models (as might be produced by a CAD system) and does not affect the appearance of the object. However, it would be of less utility for models with high curvature because these will contain fewer instances of large co-planar areas.

2.2.1.2 Data Reduction Scheme for Triangulated Surfaces (Hamann)

Hamann's (1994) algorithm is similar in concept to the previous scheme in that it also attempts to simplify nearly co-planar regions of polygons. This technique works by first assigning a weight to each triangle (calculated by averaging the curvature values at each vertex). Then the process iteratively removes the triangle of lowest weight from the mesh (i.e. least average curvature) and optimally reconstructs the local region. This is done by replacing the selected triangle with a single point and removing all of the triangle's neighbouring polygons. This new vertex is then used as the centre point for a triangulation process.

The user can control the reduction process by specifying a percentage of triangles to be removed from the model. The simplification process then halts when this target is reached, or when it can remove no further triangles due to local topology constraints. The principal advantage of curvature-based reductions is that they remove nearly planar surfaces which do not affect the overall form of an object and hence these techniques provide good shape constancy between degraded models.

2.2.1.3 Decimation of Triangle Meshes (Schroeder *et al.*)

Thus far we have looked at techniques that remove selected polygons from a model. By comparison, the decimation algorithm reported by Schroeder *et al.* (1992) removes selected vertices from the mesh. In their system, every vertex is analysed for possible removal by calculating its distance from the average plane of its adjacent vertices. If this distance is less than a user specified threshold, then the vertex—and every polygon using it—are deleted from the object description. A local triangulation scheme is then used to patch up any resulting holes. This process is repeated until a specified percentage reduction has been achieved, or no further vertices meet the decimation criterion.

Schroeder *et al.* illustrate the decimation algorithm on a number of terrain and volume data. They state that the technique performs particularly well on polygon models converted from volume data using the Marching Cubes algorithm (Lorensen and Cline, 1987).

2.2.1.4 Hierarchical Geometric Approximations (Varshney)

The geometry removal algorithm devised by Varshney (1994) introduces the notion of 'offset surfaces'. These are surfaces which are formed by extending the position of each vertex in the mesh by its normal scaled by a user supplied error tolerance. Two offset surfaces are used: one on the outside of the model and one on the inside (using positive and negative values of the error tolerance respectively).

Once the two offset surfaces have been calculated, the algorithm attempts to generate optimal triangles that lie completely within the offset surfaces and use only vertices from the original model. Varshney shows that this problem is NP–complete and subsequently presents a greedy algorithm alternative. This works by selecting the triangle that covers the most vertices in the original mesh, whilst preserving the local topology.

This technique presents a novel and intriguing means to limit the extent of a reduction. Erikson (1996) notes that the original algorithm could be enhanced by allowing the user to vary the error tolerance across the surface in order to retain visually important features. Cohen *et al*. (1996) generalise the concept of offset surfaces to produce 'simplification envelopes': a generic framework for restricting the effect of a reduction, within which a large group of polygon simplification algorithms can run.

2.2.2 Sampling

2.2.2.1 Voxel Based Object Simplification (He *et al.*)

The sampling algorithm presented by He *et al*. (1995) simplifies a polygon model by essentially converting it into a **voxel** description of arbitrary resolution, and then converting this back into a polygon description.

A 3D grid is placed over the model, segmenting it into a collection of voxels (volume elements). For each voxel, a density value is calculated by sampling all of the polygons that lie within the voxel. Then a Marching Cubes algorithm is performed on the voxel data to reconstruct a simplified polygon model. (Finally, a geometry removal algorithm is performed over the resulting mesh to reduce any redundant polygons created by the conversion.)

Changing the granularity of the 3D grid, i.e. the volume of the voxels, will affect the degree of simplification: larger voxels will generally incur a greater reduction ratio. This technique assumes that the original model is a closed shape (it has a distinct inside and outside) and does not attempt to preserve the object's topology. As a result, it can produce greatly simplified results.

2.2.2.2 Multiresolution 3D Approximations (Rossignac and Borrel)

The approach used by Rossignac and Borrel (1992) was to segment the model into a number of sub-volumes using a 3D grid. Then all vertices which exist within each sub-volume are collapsed to a single averaged position. This can produce a number of degenerate polygons that collapse to an edge or a single point—these polygons are simply removed from the model.

This technique is very efficient and relatively simple to implement. However the results can be extremely coarse and often drastically degrade the form of the model. One refinement which can reduce this effect is the introduction of a tolerance factor. This essentially limits the collapsing of vertices to within a certain distance of the averaged vertex (MultiGen, Inc., 1994). Also, vertices that lie on the convex hull of the object can be preserved in an attempt to retain the general form of the object.

2.2.2.3 Mesh Optimisation (Hoppe *et al.*)

Hoppe *et al.* (1993) present a triangular mesh simplification process which was based upon their surface reconstruction work (Hoppe *et al.*, 1992). This technique introduces the concept of an energy function to model the opposing factors of polygon reduction and similarity to the original topography. The energy function is used to provide a measure of deviance between the original mesh and the simplified version. This is then minimised to find an optimal distribution of vertices for any particular instantiation of the energy function.

Hoppe *et al.* note that their mesh optimisation technique successfully distributes vertices in relation to surface curvature (i.e. areas of high curvature are densely coded with vertices: whereas relatively flat regions contain fewer vertices), thus providing a high degree of shape constancy between model approximations.

2.2.2.4 Re-Tiling Polygon Surfaces (Turk)

The re-tiling technique formulated by Turk (1992) optimises a polygon mesh by introducing new vertices to the mesh and then discarding the old vertices to form a new representation. This involves creating an initial triangulation of the surface with a user-defined number of vertices. These new points are pseudo-randomly positioned in the planes of the existing polygons and then successively repelled by their neighbours in order to create a uniform distribution. Once this relaxation process has converged, the old vertices are removed one by one and the surface is locally re-tiled at each step in order to retain the topography of the original surface.

Turk notes that this technique works best for curved surfaces (such as the iso-surfaces from medical data or molecular graphics) and that it is less suited to angular entities such as buildings, furniture, or machine parts.

2.2.2.5 Simple Replacements for Groups of Objects (Sewell)

Sewell (1995) suggests an intriguing approach for approximating certain classes of objects. He developed a mechanism which computes a simple replacement primitive for a complex grouping. For example, his initial system analysed a model and attempted to approximate parts of it with spheres, boxes or ellipsoids (this technique has perceptual grounding in the concept of **geons**, proposed by Biederman, 1987. This theory suggests that human object recognition is based upon identifying a small number of primitive shapes within an object).

Sewell comments that this approach can produce substantial complexity reductions because it can decompose clusters of objects which most other techniques would treat as separate entities. However, he notes that the technique can generate various visual artifacts as a result of this.

2.2.2.6 The GENIE Level of Detail Generation System (Kemeny)

The SCANeR project is a networked driving simulator, mentioned previously in Section 2.1.1. The system makes use of an interactive software package called GENIE which generates various level of detail models of a desired object. These are used in order to perform distance level of detail switching, i.e. the level of detail of an object is selected depending upon its distance from the user.

The GENIE system works by creating a 2D grid and then projecting all the vertices of an object onto that grid. The system then attempts to reduce this representation so that only one polygon occupies any one grid cell (or **rexel**). By altering the scale of the grid cells based upon the desired resolution and the observation distance, it is possible to create a range of level of detail models for the simulation.

For the particular application area, the GENIE system offers an efficient model of level of detail generation. However, it utilises a viewpoint dependent algorithm which assumes that the model does not rotate around its axes. This assumption is perfectly valid for a driving simulator in which the scenery is static and simply grows in size as the user approaches it; however, for a more general virtual environment, a viewpoint invariant algorithm is required.

2.2.3 Adaptive Subdivision

2.2.3.1 Simplification of Objects Rendered by Polygonal Approximations (DeHaemer, Jr. and Zyda)

All of the geometry removal schemes that we have already looked at could be described as top-down, in that they take the original polygon description and attempt to remove detail from it. By comparison, adaptive subdivision is a bottom-up approach: it begins with a crude approximation of the object and then recursively refines this by subdividing the model where it varies most from the original mesh.

Schmitt *et al.* (1986) were the first to suggest this technique. They used surfaces composed of bicubic Bernstein-Bézier patches and developed a metric to model the 'closeness' of an approximation to the original data. Based upon this method, DeHaemer, Jr. and Zyda (1991) developed a similar technique for certain classes of polygon meshes. Their system begins with a single quadrilateral polygon approximating the entire model. This is then recursively subdivided until all polygons are within a user specified distance from the original mesh. The subdivision process can often create gaps between adjoining polygons, so these are filled with additional polygons.

DeHaemer, Jr. and Zyda used this technique to simplify complex 3D range-data obtained from the laser scanning of real-world objects. The algorithm takes advantage of the inherent regularity of these mesh data and as a result its application to arbitrary 3D models is limited.

2.2.3.2 Multiresolution Analysis of Arbitrary Meshes (Eck *et al.*)

Building on the initial work of Lounsbery *et al.* (1994), Eck *et al.* (1995) created a simplification algorithm for arbitrary polygon meshes based upon wavelet theory. Wavelets are a means of hierarchically decomposing a function so that it can be described as a coarse general form, augmented by a series of local correction terms that capture detail at different scales. Their applicability to polygon reduction is illustrated by Stollnitz *et al.* (1995a & b) who lucidly explain how a series of approximations to an original object can be formed by omitting a number of these small detail terms (wavelet coefficients) when rebuilding the model.

Eck *et al*. (1995) note that their algorithm offers potential benefits in a number of application areas, including: object compression, LOD, and multiresolution editing. However from our perspective, the principal advantage of wavelet based systems is that they can produce smooth transitions between different levels of detail, because the wavelet coefficients can be added continuously.

2.2.3.3 Superfaces (Kalvin and Taylor)

Kalvin and Taylor's (1994) algorithm attempts to merge all of the polygons in a model into a number of conglomerate 'superfaces' and then subdivides these until they meet a user specified error bound.

The first stage is to generate all of the superfaces. This is done by selecting a seed polygon and then successively merging all of the adjacent polygons that satisfy three criteria: the user's error bound, good aspect ratio and a local topology constraint. Once all the superfaces have been generated in this way, the algorithm attempts to collapse the connecting edges between neighbouring superfaces to a single edge so that their boundaries are more regular. If this edge straightening processes produces edges that violate the error bound, then they are recursively subdivided until the criterion is met. Finally, each superface is triangulated.

The superfaces solution can be rapidly computed and guarantees that every vertex of the simplified mesh is within a specified distance of the original mesh. However there are still a number of unresolved problems, such as how to deal with holes in a superface.

2.2.4 Discussion of LOD Generation

2.2.4.1 Identification of Factors Concerning Polygon Simplification

We can see from the above analysis that a wide range of techniques has been developed to simplify polygonal models of objects. However, it may also be evident that there are a number of factors to be considered and balanced when we talk about any individual technique. Some of these are discussed below; while Table 2.1 attempts to characterise each of the above techniques in terms of these factors.

- **<u>Topology Preserving or Simplifying</u>** : topology in the current context refers to the geometric structure of an object. We usually talk about **local topology** (the geometric structure surrounding a certain polygon or vertex), or **global topology** (the geometric structure of the entire model). An algorithm which preserves topology therefore retains the prominent features in an object such as holes or spikes; whereas a topology simplifying algorithm would smooth over these features and therefore destroy some of the object's structural content.

  To use Erikson's (1996) example: a topology preserving algorithm would not produce good results on a model representing a slice of Swiss cheese because it could not simplify the holes without violating local topology. However, a topology simplifying algorithm could smooth over these features, leaving a solid slice with no holes.

  As a result, topology simplifying algorithms can generally produce greater reductions because they have fewer constraints. However, as a consequence these models can often appear highly degenerate when compared to the original object.

- **<u>Polygonal Dataset Constraints</u>** : we find that very few algorithms can operate on all classes of polygonal datasets. (In fact, Luebke, 1996, suggests that a perfect and generic simplification algorithm is unlikely to appear). Instead, most algorithms only accept models in a certain format. Normally it is required that polygon data are provided as triangular meshes, with no coincident triangles, and no edges shared by more than two triangles, etc.

  Therefore if a particular algorithm will not accept a certain model, then the polygon data must be converted into an acceptable format, or an alternative algorithm must be adopted.

- **<u>Vertex List Coherency</u>** : the final consideration which we will present here is whether the vertices in the simplified model are a sub-set of those in the original. A number of algorithms generate new vertices and hence introduce additional entries into the vertex list. This may be of little concern for most applications, although models which share a common vertex list could feasibly make more optimal usage of storage and memory resources.

Author(s) & Date	Algorithm Genus	Polygon Constr.	Vertex Subset?	Preserve Topology?
DeHaemer, Jr. and Zyda (1991)	A. S.	quads	✔	✔
Turk (1992)	S.	tris	✘	✔
Schroeder *et al*. (1992)	G. R.	tris	✔	✔
Rossignac and Borrel (1992)	S.	polys	✘	✘
Kemeny (1993)	S.	polys	✔	✘
Hoppe *et al*. (1993)	S.	tris	✘	✔
Hinker and Hansen (1993)	G. R.	tris	✔	✔
Hamann (1994)	G. R.	tris	✘	✔
Kalvin and Taylor (1994)	A. S.	tris	✔	✔
Varshney (1994)	G. R.	tris	✔	✔
Eck *et al*. (1995)	A. S.	tris	✘	✔
He *et al*. (1995)	S.	tris	✘	✘
Sewell (1995)	S.	polys	✘	✘

Table 2.1: *A chronology of polygon simplification techniques. Table headings include: the genus of algorithm (G. R. = geometry removal, S. = sampling, A. S. = adaptive subdivision), the type of polygons which the algorithm operates on (triangles, quadrilaterals, or arbitrary polygons), whether the new vertices are a subset of the original, and whether the algorithm preserves local topology.*

2.2.4.2 A Mandate for Polygon Simplification in Virtual Environments

After considering the above techniques and factors, there are still some areas of polygon reduction that have received surprisingly little attention; yet they are extremely important for the effective use of level of detail in virtual environments. Principal among these deficiencies are:

- **Lack of Surface Material Consideration** : modern graphics systems do not usually display objects as simple wireframe models or bland single-coloured entities. Instead, an object will normally contain various polygons of different colours and may also included texture mapped polygons. Practically all of the reviewed polygon reduction techniques have been concerned solely with the simplification of geometry, and do not take into consideration the surface properties of a model.

  For example: consider two co-planar and adjacent polygons that we wish to merge into a single description. If one of these polygons is yellow and the other one contains a brick effect texture map, then what surface properties should the simplified polygon have? Or indeed, should we even consider simplifying these two polygons if their surface properties are so disparate?

- **Lack of Perceptual Predictability** : we have already noted that one of the problems associated with LOD is the popping effect; where switching between different levels of detail incurs a noticeable flicker. This is due primarily to the fact that the simplified models have been arbitrarily reduced and so there is little basis to formally define the LOD thresholds. In essence, we have no quantitative measure for the effect that a simplification will have on the user's perception of that object.

The lack of work in these areas can be attributed to the fact that most simplification algorithms have been developed for applications such as medical imaging. These domains are usually concerned with the accurate visualisation of 3D monochromatic structures. In contrast, the field of real-time computer graphics utilises full-colour scenes and is more concerned with performance than fidelity of the VE. For example, reference to Table 2.1 will confirm that most polygon simplification algorithms are topology preserving, but this is not an important issue for our purposes: if an oak tree is displayed such that it occupies only three pixels of the screen, then we do not care if every leaf is accurately represented. We would prefer a topology simplifying algorithm that can smooth over individual leaves and thus greatly reduce the complexity of the model.

We can therefore attest that most of the current polygon simplification techniques do not meet all of the requirements demanded by our application. In an attempt to rectify this situation, we will suggest three base criteria for polygon simplification algorithms to ensure their suitability for a VR system that modulates LOD based upon perceptual determinations:

1. The effect of the simplification should be perceptually predictable.
2. The surface properties of the model should be taken into consideration.
3. A topology simplifying algorithm should be advocated.

Criteria 2 and 3 are general requirements for the use of LOD in a VR system. Criterion 1 is a further requirement to support the implementation of perceptually modulated LOD. It is not deemed necessary to rigidly specify any other factors, such as the particular algorithm genus or whether vertex coherency is desired. However, it is worth noting that all topology simplifying techniques to date have been sampling algorithms. This might lend credence to the corollary of criterion 3 stating that a sampling polygon simplification algorithm should be advocated for the use of LOD in VR systems.

2.3 Models of Visual Perception

Having expounded the computer graphics issues, we will now turn our attention to the field of visual perception. The following sections will introduce the techniques which have been utilised to measure the efficacy and limitations of the human visual system. These attempt to answer the question of how much information the eye can actually resolve. Before we investigate this however, we will introduce a couple of the more prominent theories of visual perception which have existed this century.

2.3.1 A Review of Visual Perception Theories

2.3.1.1 Gestalt Psychology

For most of the first half of this century, one of the most prevalent theories of visual perception was that of **Gestalt psychology**. Formed in 1912, the advocates of this school believed that overall structure is a more important determinant of form perception than the individual components of an image (the word Gestalt is German for ‘form’). A number of the more prominent Gestalt psychologists (including Max Wertheimer, Wolfgang Köhler, and Kurt Koffka) were responsible for the formulation of the **Gestalt principles of organisation**. These rules attempt to encapsulate the principal factors that contribute towards a group of items being perceived as a single entity. A number of the more important of these are paraphrased below:

1. **<u>Proximity</u>** : items which are spatially close to one another tend to form a perceptual entity.

2. **Similarity** : similar items tend to be grouped together, e.g. regions of similar lightness, orientation, or size.
3. **Closure** : items tend to coalesce to form a closed, solid mass.
4. **Good Continuation** : items that are potentially connected by a straight line or smooth curve appear to group together.

Gestalt psychologists also developed a neural model of vision to support their theories. They believed that the electrical activity in the brain would correspond directly to the image formed on the retina: that there would in effect be a pattern of neural stimulation resembling the pattern of the retinal image (this theory was termed **isomorphism**). However, more recent neurophysiological studies have found these conjectures to be incorrect. A new approach was therefore required to explain the function of the visual system.

2.3.1.2 The Multichannel Model

The most widely accepted contemporary theory of spatial vision is that of the **multichannel model**. Developed from the work of Enroth-Cugell and Robson (1966) and Campbell and Robson (1968), this theory essentially proposes that the visual system processes the retinal image simultaneously at several different spatial scales.

It can be noted that most naturally occurring scenes contain visual information at a number of different scales. For example, if we consider the case of a forest: the outline of all the trees provides a coarse degree of detail; we could then focus on each individual tree, or we could concentrate on the finer detail of the leaves on a single tree. The multichannel model suggests that the visual system extracts all of these different scales of information in a scene simultaneously, and that these are later combined by the higher vision processes to assemble our final percept for the particular scene.

This theory can be substantiated by our knowledge of the neural design of the visual system. As we have seen in the preceding chapter, the human visual system is sensitive to gradients of light intensity rather than absolute levels of illumination. In real world terms this implies that the visual system is sensitive to edges. We have also seen that the size of a neuron's receptive field defines the size of stimulus to which it is maximally sensitive. At any point in the visual system, there is a collection of cells with a range of receptive field sizes: and therefore a range of detail which can be detected. In this respect, we can define a **channel** as simply a class of neurons with a certain receptive field size.

There is still contention over precisely how many channels exist within the visual system (Heeley, 1991; Caelli and Moraglia, 1985; Harvey and Gervais, 1981; Wilson and Bergen, 1979). However, the major point to bear in mind is that the multichannel model predicts that information is analysed independently by a number of parallel channels, each of which is tuned to a particular level of detail.

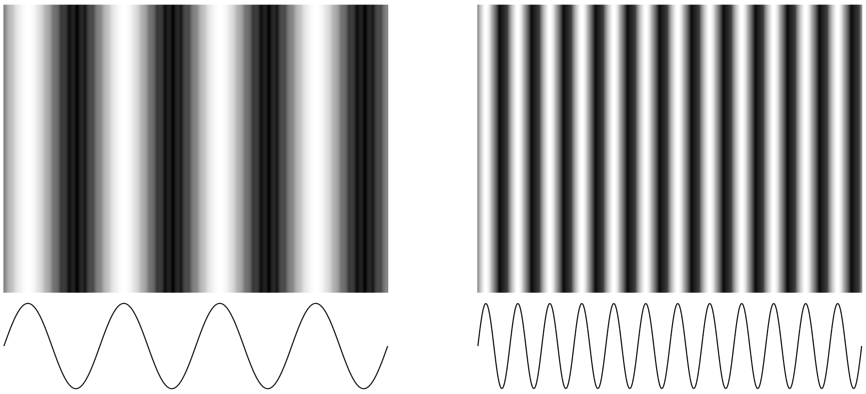
2.3.2 Measuring Visual Acuity

2.3.2.1 Contrast Gratings and Spatial Frequency

The human ability to perceive detail is determined by the relative size and contrast of a stimulus (Campbell and Robson, 1968). This faculty has been extensively analysed by vision researchers for the past several decades. Since the pioneering work of Schade (1956), the most common experimental device for accurately measuring a subject's visual acuity is the **contrast grating**. This is simply a pattern where contrast is varied sinusoidally across the display, producing a series of alternating light and dark vertical bars (see Figure 2.2).

There are two principal independent factors which affect the appearance of a contrast grating: its **contrast** and **spatial frequency**. Contrast is simply a measure of the luminance difference between adjacent light and dark bars; whereas spatial frequency is a measure of the spacing between bars, defined in units of contrast cycles per degree of visual field (c/deg). For example, a high spatial frequency implies a short distance between adjacent bars and hence represents a stimulus of high detail.

For a number of different contrast gratings, the limits of human vision can be investigated and recorded in terms of these two parameters. This is normally done by allowing the subject to vary the contrast of a grating until it is deemed to be at threshold, i.e. they can no longer resolve discrete bars (Lamming, 1991a). Much of this initial work was performed by the late Fergus Campbell and his colleagues (e.g. Campbell and Green, 1965; Campbell and Gubisch, 1966; Camp-



(a) **(b)**

Figure 2.2: *An illustration of two contrast gratings displaying: (a) a low and, (b) a higher spatial frequency. The curve below each of the gratings shows the sinusoidal nature of the intensity distribution. If grating (a) was positioned to occupy 1 deg of visual arc, then it would have a spatial frequency of 4 c/deg.*

bell and Robson, 1968; Blakemore and Campbell, 1969).

In this fashion, it has been empirically confirmed that our ability to resolve detail varies in relation to the orientation of a contrast grating (Campbell *et al.*, 1966), its velocity across the retina (Kelly, 1979), the degree to which it is placed in our peripheral field (Rovamo and Virsu, 1979), and the level of background illumination (Kelly, 1975). The phase of a contrast grating has no effect on its detectability (Lamming, 1991c).

2.3.2.2 The Contrast Sensitivity Function

The **contrast sensitivity function** (CSF) is essentially a graph of the results from a series of contrast grating tests. It illustrates the threshold of vision for a single observer at a number of spatial frequencies. Accordingly, it is often stated that the CSF describes a subject's **window of visibility**; because the region below the curve represents combinations of spatial frequency and contrast that were visible to the subject.

From Figure 2.3 we can observe that our vision system exhibits a peak sensitivity (around 3 c/deg) and that our ability to resolve detail drops off after this peak until there is a point beyond which we can resolve no further detail. For example (based upon the CSF in Figure 2.3), if we presented the subject with a stimulus of 100 c/deg, then they would simply not be able to see it: it would be invisible to their eye.

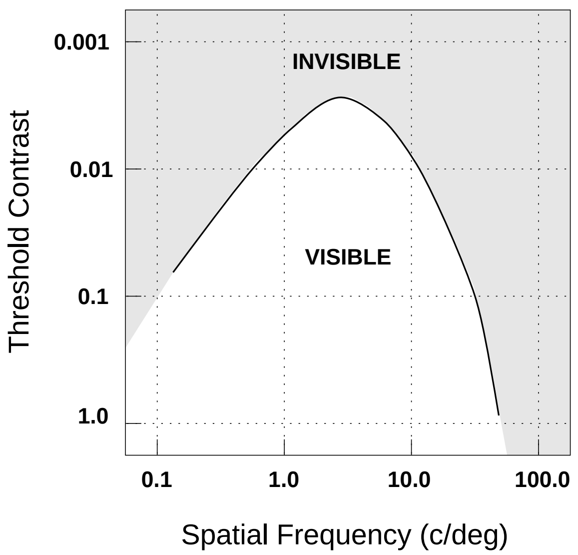


Figure 2.3: *An example contrast sensitivity function for static detail. This represents the size-sensitivity of the human visual system in relation to the contrast of the stimulus.*

The curve in Figure 2.3 is for static detail presented at the observer's fovea. If we look at the corresponding curves for moving gratings or eccentric gratings, then we find that the CSF shifts towards the ordinate in both cases (Nakayama, 1990; Koenderink *et al.*, 1978a). Effectively, this means that we can perceive fewer high spatial frequencies under these situations: less high detail.

Given a mathematical equation to model the shape of the CSF under various conditions, we can compute the highest spatial frequency that an observer should be able to see. This provides us with our metric (spatial frequency) and model (contrast sensitivity) to quantify the degree of detail that the user of a VR system can see.

As we have now gained a basic grasp of the background vision material, we can attempt a more formal definition for some of the terms that we have been using thus far:

Threshold Contrast : the level of contrast (0–1.0) above which a particular stimulus becomes visible. For example, a low threshold contrast would mean that the stimulus is visible under low contrast conditions, and all contrasts higher than this; whereas a high value of threshold contrast implies that a stimulus must be presented at high contrast before it becomes visible.

Contrast Sensitivity : the term contrast sensitivity is defined as the reciprocal of threshold contrast, i.e.

Contrast Sensitivity = 1 / Threshold Contrast.

Visual Acuity : a measure of the smallest detail that a person can resolve. This is only a measure of size and does not take into consideration the contrast of a target. Visual acuity is therefore normally assessed under optimal illumination conditions, e.g. black letters on a white background under bright lighting.

2.3.3 Applying Vision Results to Computer Graphics

Simple contrast gratings bear little resemblance to the images displayed by a computer graphics system. We must therefore illustrate the applicability of these devices to our domain. Accordingly, we can note that a contrast grating differs from a computer-generated image in the following respects:

1. **Modulation** : a contrast grating presents a sine-wave distribution of intensity. However, computer-generated images rarely contain perfectly harmonic features, e.g. a simple flat-shaded object would present a square-wave distribution of intensity across the display. We must therefore consider the applicability of a sine-wave grating to square-wave (and other) gratings.

2. **Complexity** : most computer-generated images involve complex changes of intensity across the display. As a contrast grating is a simple harmonic pattern, we must consider the visibility of a complex intensity waveform in terms of the simple harmonic case.

3. **Periodicity** : a contrast grating is a periodic sine-wave pattern, normally containing several complete cycles of contrast. Such patterns do not commonly occur in computer-generated scenes. Instead we will often be concerned with the visibility of an aperiodic region of detail. We must therefore investigate whether periodicity can affect the visibility of a particular

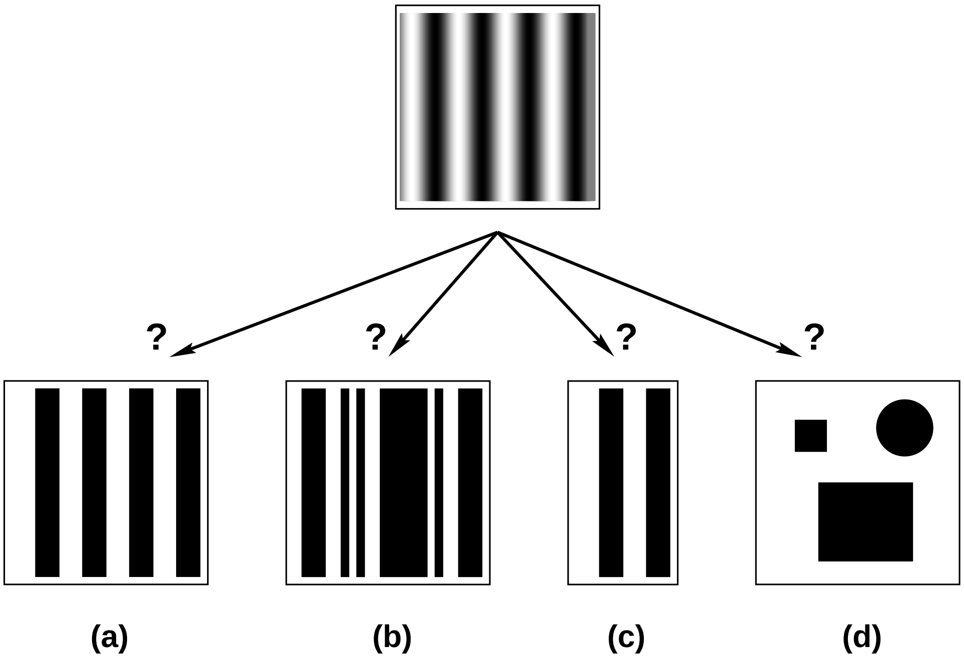


Figure 2.4: *How does a simple 1D harmonic contrast grating relate to: a) square-wave gratings, b) complex gratings, c) gratings of different periodicity, and d) 2D gratings?.*

spatial frequency (i.e. the number of contrast cycles present in the stimulus).

4. **Dimension** : a contrast grating varies over only one dimension (1D), whereas a computer image is obviously two dimensional (2D). We must therefore investigate how a 2D image can be described in terms of spatial frequency.

5. **Chromaticity** : computer-generated imagery is often presented in colour rather than greyscale. We therefore have to investigate whether the detection threshold of an achromatic pattern such as a contrast grating can be applied to a chromatic display.

Figure 2.4 illustrates each of these points (except for the chromaticity case because this dissertation is not presented in colour). The subsequent sections will take each of these factors in turn and assess the relevant implications for computer graphics systems.

2.3.3.1 Modulation Effects on Feature Detection

In their influential paper, Campbell and Robson (1968) showed how contrast gratings other than sine-wave gratings can be related to the simple harmonic case. To illustrate this, we will examine how the visibility of a square-wave grating relates to that of a sine-wave grating (as in Figure 2.4(a)). This is a particularly pertinent example for computer graphics displays because a flat-shaded image will essentially contain a square wave distribution of intensity.

We can observe from Figure 2.5 that above the peak frequency, the amplitude of the square-wave CSF is largely determined by that of the sine-wave curve; and that below the peak frequency, the square-wave curve levels off at a certain threshold contrast. This behaviour can be predicted by Fourier theory: a square wave, $\Pi(x)$, of unit amplitude and period X can be described as the sum of an infinite series of harmonics,

$$\Pi(x) = \frac{4}{\pi}\left(\sin\frac{2\pi x}{X} + \frac{1}{3}\sin 3\frac{2\pi x}{X} + \frac{1}{5}\sin 5\frac{2\pi x}{X} + \ldots\right). \tag{2.6}$$

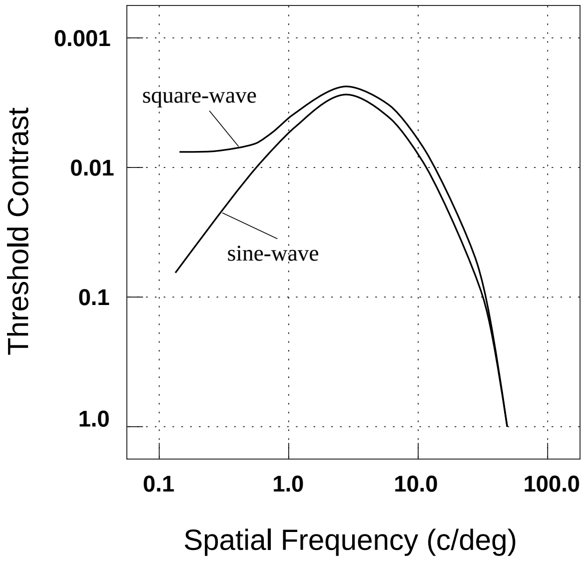


Figure 2.5: *Contrast sensitivity compared for sine-wave gratings (lower curve) and square-wave gratings (upper curve). Adapted from results in Campbell and Robson (1968).*

The amplitude of the first harmonic (the **fundamental**) component is therefore $4/\pi$ that of the square-wave amplitude; with the third, fifth and higher harmonics having relative amplitudes of $4/3\pi$, $4/5\pi$, etc. (The even harmonics all have zero amplitude.)

Campbell and Robson hypothesised that beyond the peak frequency, the visibility of a square-wave grating would be characterised by the amplitude of its fundamental component, i.e. the amplitude of the CSF for a square-wave would be $4/\pi$ that of a sine-wave grating of equal contrast. Below the peak frequency, the higher harmonics of the square-wave become progressively visible and so the visibility of the grating would no longer be defined solely by its fundamental component. This elegant hypothesis was supported by their empirical results, illustrated in Figure 2.5.

We can therefore see that the CSF for a square-wave grating can be derived from knowledge of the harmonic CSF. Campbell and Robson showed that similar results hold for rectangular, triangular, and other modulations.

To summarise: for frequencies beyond the peak frequency, the modulation of a grating has no significant effect on the visibility of that grating (e.g. Campbell and Robson noted that a square-wave grating is indistinguishable from a sine-wave grating until the third harmonic reaches its own threshold). That is, for the upper regions of the CSF we can analyse a stimulus by only referring to its fundamental component (Lamming, 1991c).

2.3.3.2 Complexity Effects on Feature Detection

Campbell and Robson also performed experiments with compound contrast gratings in order to investigate how the visibility of these is related to that of their component harmonic gratings (Sekuler and Blake, 1994). They found that the appearance of a compound grating is characterised by the independent contributions from each of the harmonic components. Their results showed that if a compound grating is displayed such that some of its high frequency components are below threshold, then these features will not be visible in the compound grating and can be removed without any perceivable change being made to the grating.

This was one of the major contributing results in the development of the current multichannel model of visual perception. Its implication for our purposes is that the visibility of the component detail in a complex image can be assessed independently in terms of the simple CSF threshold data.

2.3.3.3 Periodicity Effects on Feature Detection

Coltman and Anderson (1960) showed that the number of cycles in a sine-wave pattern can have a substantial effect on contrast sensitivity. This work was later re-examined by Nachmias (1968) who was concerned with the visibility of square-wave patterns at low spatial frequencies (below 10 c/deg). He found a consistent 60% reduction in sensitivity for single cycle patterns compared to full gratings.

Contemporaneously, Campbell *et al*. (1969) investigated this phenomenon for the high spatial frequency range. They found that the highest detectable frequency for one cycle gratings is considerably higher (e.g. $>$ 100 c/deg) than for an extended sinusoid grating. Their results are illustrated in Figure 2.6. From these data we can observe that at lower frequencies, our sensitivity is reduced for single cycle gratings. However, our visual acuity is extended for these gratings compared to full gratings.

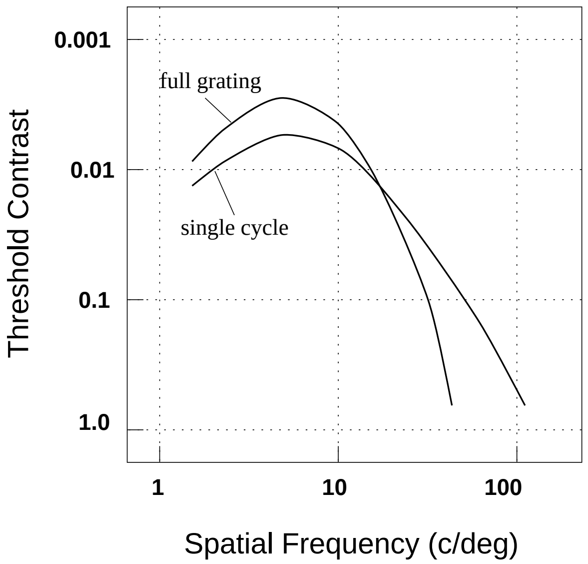


Figure 2.6: *Contrast sensitivity compared for full gratings and single cycle (double bar) gratings. Adapted from Campbell et al. (1969).*

The implication for our work is that the standard CSF data for extended contrast gratings may underestimate the contrast sensitivity of a user to detail on a computer screen. This is because a feature in a computer-generated image will usually represent a half-cycle stimulus, i.e. a single peak (or trough) intensity with respect to the surrounding region. However, Campbell *et al*. (1969) suggest that the visibility of an aperiodic pattern can theoretically be predicted from that of a sinusoidal grating, and that a simple linear relationship may exist.

2.3.3.4 Dimension Effects on Feature Detection

Spatial frequency, as we have defined it so far, is an inherently 1D measure: it describes the intensity variation over a single cross-section of a display. In order to describe a 2D feature using this measure we introduce an orientation parameter for each frequency. A 2D feature is then described by the set of spatial frequencies at all angles $(0..180^{\circ})$. Figure 2.7 illustrates this concept by presenting a number of spatial frequencies over a range of orientations.

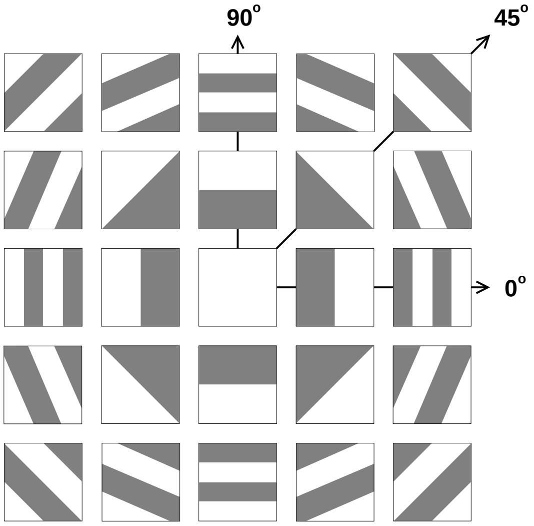


Figure 2.7: *An illustration of 1 cycle and 2 cycle spatial frequencies at various orientations.*

For example, the circle in Figure 2.4(d) will have a horizontal spatial frequency (i.e. an orientation of 0°), a vertical spatial frequency (90° orientation), and frequencies at all intermediate angles. (Because the diameter of a circle is the same for every orientation, all of these spatial frequencies will be equal in this particular case.) To illustrate this point further, consider a long thin object such as a street lamp post. Such an object is considerably taller than it is wide. This object would therefore have a very low vertical frequency (i.e. long vertical distance) and a comparatively high horizontal frequency (i.e. short horizontal distance).

2.3.3.5 Chromaticity Effects on Feature Detection

A contrast grating is a greyscale pattern (achromatic), but the images produced by a VR system are invariably colour (chromatic). We must therefore ask if we would lose any accuracy by employing achromatic threshold data to our task; and whether we should consider applying colour contrast data instead.

We know that the achromatic channel is far more effective than the chromatic channels for processing shape (Mullen, 1985), motion (Anstis and Cavanagh, 1983), and stereoscopic depth (Gregory, 1977). For example, Campbell and Gubisch (1966) identified the upper spatial limit of the luminance channel as $\sim$60 c/deg; whereas Mullen (1985) found that the red/green upper limit is only $\sim$12 c/deg: i.e. the achromatic channel can resolve substantially smaller features than the chromatic channels. (One reason for this could be due to the fact that a single chromatic response depends upon the integration and comparison of a local group of wavelength-specific cones.)

Gregory (1977) believes that luminance information is used to extract the detail in a scene; while colour information is used to 'fill in the gaps' (e.g. for identification and recognition purposes, to process the material properties of a surface, etc.). As an example, he notes that face perception is destroyed when only colour information available, i.e. at isoluminance (Gregory, 1993). Hubel and Wiesel (1962) also suggest that colour makes very little contribution to spatial and temporal vision.

It is evident that colour is important for suprathreshold vision (Cavanagh, 1991), however we can see that it plays a far inferior rôle to luminance for threshold vision. Additionally, as we are also interested in the perception of objects in the user's peripheral field, we should note that the photoreceptors in the extrafoveal region are predominantly rods, which are achromatic sensors.

In conclusion therefore, not only are the achromatic CSFs applicable to our task, but they are indeed a more accurate measure of visual acuity than if we were to attempt to use chromatic contrast thresholds.

2.3.4 Discussion of the Vision Material

2.3.4.1 Assessing the Relevant Visual Perception Work

We have seen that our ability to resolve spatial detail is dependent upon the contrast and relative size (spatial frequency) of a stimulus. We have subsequently looked at two models of stimulus detection: contrast sensitivity and visual acuity. We have concentrated on contrast sensitivity for the following reasons:

1. Of these two, contrast sensitivity provides a more complete model because it takes into consideration the contrast of a stimulus, whereas visual acuity is simply a measure of the smallest resolvable size under ideal illumination conditions.

2. The literature pertaining to contrast sensitivity is considerably more extensive than that for visual acuity. The latter remains a measure of static detail viewed under foveal conditions; whereas much research has been performed to describe our ability to resolve motive and eccentric targets using contrast sensitivity.

3. Visual acuity tends to be described in more computationally esoteric terms than contrast sensitivity. The most common measure of visual acuity is the **Snellen fraction**; named after the Dutch doctor, Hermann Snellen, who introduced the technique. A Snellen fraction of $20/n$ is defined as the acuity at which two objects, which subtend 1 min of arc at n ft, can be perceived as separate at 20 ft (Tipton, 1984). Therefore a person with 20/20 vision is classed as normal, and a person with 20/40 vision can only see a stimulus from 20 ft that a normal person can see from 40 ft.

It is interesting to note that given an observer's contrast sensitivity, we can derive their visual acuity in terms of spatial frequency. This is simply the upper limit of detection, i.e. the rightmost point where the CSF meets the abscissa. Figure 2.8 illustrates this relationship.

One important point to bear in mind throughout this dissertation is that our vision system is not a simple, predictable automaton. There are many factors that affect our ability to perceive detail, and indeed everybody's visual system is slightly different. (We shall discuss this issue of subjectivity in greater depth in Chapter 6.) Finding a model which can precisely encapsulate what every

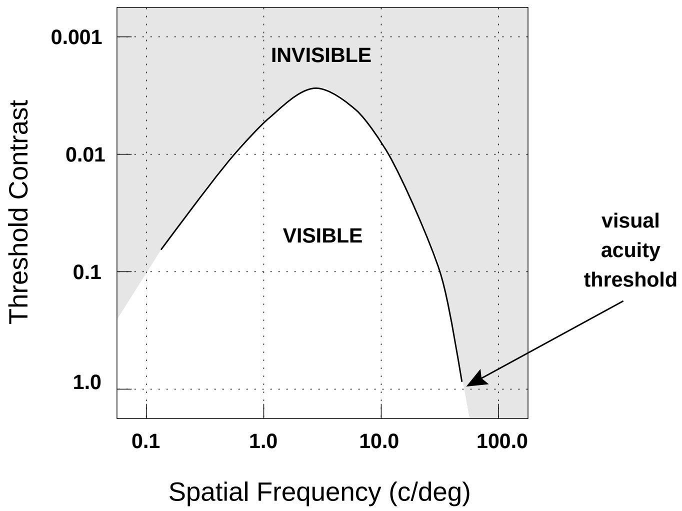


Figure 2.8: *A portrayal of how visual acuity can be described in terms of spatial frequency given a subject's contrast sensitivity function.*

person can see under every circumstance would be a Heraclean task. Instead, vision scientists refer to a **standard observer**. This is simply a notional 'average' human for whom we can develop general models of perception (normally under prescribed conditions) which are applicable to most of the population.

It is also worth noting that there is still no adequate general theory of visual perception. As yet, we do not fully understand how the brain performs the non-trivial task of vision: taking the optic data fed to it from the eyes and identifying discrete objects therein. However, the processes involved in early vision are well understood, i.e. how light is sensed by the retina, how it is filtered through various synaptic networks, and how it is transported to the visual cortex. These functions determine the raw data that are available to the ensuing vision processes and have been studied thoroughly for many years. As a result we have a solid understanding and appreciation for the early stages of vision, encapsulated by the multichannel model.

In this thesis we are concerned with the visibility of features in a scene (so that we can decide when detail can be reduced without affecting the user's percept of the VE). As such, we are only interested in the threshold efficiency of our visual system to resolve stimuli. We are not concerned with the higher vision processes such as object identification, classification, or recognition. Our work should have no bearing or effect on these functions because we are attempting to remove visual detail from the scene which would not be available to them in the first place.

2.3.4.2 Mandate for the Use of Vision Material in Computer Graphics

We have already shown that a model of contrast sensitivity can theoretically be applied to a computer graphics system (Section 2.3.3). Let us now examine more closely how this can be achieved.

In order to optimise spatial detail based upon any model of visual perception we require a computer system with the ability to describe the perceptual content of a scene in terms of this model, and also to quantify the efficacy of an observer's perception in terms of this model. With these two facilities, the computer system can judge which detail a user can and cannot see in a computer-generated scene. For our model of contrast sensitivity, we therefore require the following:

1. A machine-computable mechanism to concisely describe any model in a VE in terms of its component spatial frequencies (c/deg).
2. An efficient mathematical system to predict the contrast sensitivity of a standard observer under various visual conditions (e.g. variable velocity and eccentricity).

Both of these tasks are non-trivial and deserve entire Ph.D. studies in their own right. Nonetheless, we will attempt to formulate solutions to these problems in the next chapter.

2.4 Summary

This chapter has covered a substantial volume of material; presenting and discussing the published literature in three distinct—yet related—fields.

We have seen that computer graphics researchers have already investigated reducing detail in a scene based upon the size, velocity, and eccentricity of objects. However there is currently no mechanism to find the optimal LOD to display at any time. In order to evaluate this, we would need a metric to describe the degree of spatial detail that a user can perceive in an object, and also a model of human threshold visibility defined in terms of this metric.

After looking at the visual perception background we have found two such quantities. Namely spatial frequency (as a measure of perceived detail), and contrast sensitivity (as a model of visibility). Spatial frequency is simply a measure of intensity change over the visual field; whereas contrast sensitivity describes the threshold contrast of a user for a stimulus of any spatial frequency. We therefore require a system which can compute all the relevant spatial frequencies within an object and also efficiently evaluate a user's contrast sensitivity at any point in the scene.

We have also presented an extensive review of the field of polygon simplification. This topic is essential to the field of LOD because it enables the production of different representations of an original model. We have seen that despite the wealth of published techniques, there is only partial support currently available for the use of LOD in VEs. Specifically we found that, for our particular application, we require a topology simplifying algorithm which considers the surface properties of a model and which can produce perceptually predictable results.

Through our literature survey of the above material, this chapter has highlighted a number of requirements for the development of a perceptually-based LOD system. The following chapter will therefore address each of these requirements, and attempt to formulate practical solutions to each.

Chapter 3

Development

'All that we can ever comprehend of the Universe must reside within our minds. [...] *The amazing thing is not that mental models of nature are flawed or diminutive, but that they work at all.'*

(Timothy Ferris, *The Universe and Eye*)

This chapter will present solutions to the various problems that were identified as a result of the preceding literature review. Specifically, we will develop a means to describe any object in a VE in terms of its component spatial frequencies. We will also develop a mathematical model to predict the contrast sensitivity of a standard observer to a stimulus of any velocity and eccentricity. Finally we will produce a framework for polygon simplification which is perceptually predictable and which takes into consideration the surface properties of a model (in addition to the base geometry).

3.1 A Measure for Perceived Detail

3.1.1 Where to Calculate Detail?

One of the first and most critical design decisions which we encounter when building a perceptually-based LOD system is *where* to calculate the perceptual content of an object. We have two choices: either we base our analysis on the (3D) geometric definition of an object (i.e. its polygons and vertices), or the

(2D) rendered image of that object. Both approaches have their distinct pros and cons which we can expand as follows:

Geometrical Description : here we have 3D data available to us, so we can compute the projected 2D spatial frequency profile for any orientation of the model. Also, this could feasibly be computed on-line; during the actual simulation.

However, any such algorithm would probably have to be tied quite closely to the particular rendering model chosen (e.g. polygonal, ray tracing, radiosity, etc.) because it must manipulate the internal representation of an object. Additionally, some visual effects are not described by simply looking at the geometry of a model, e.g. a texture mapped polygon may contain various intricate visual details which would be completely ignored if we only referred to the underlying geometry.

Rendered Image : here we have a 2D snapshot of an object from a single viewpoint. We would therefore have to take multiple snapshots around the object and interpolate between these in order to describe its 3D profile. We would also have to apply the computation off-line because we are required to know the spatial frequency content of a model *before* we display it: once a model has been rendered, we have already expended the computational resources which we were seeking to preserve.

The advantages however are that because we only apply our computation to the final rendered image, the method is independent of the particular rendering technique employed. Also, the rendered image more accurately and completely reflects the visual stimuli which the user is actually presented with.

Based upon the above, it is clear to the author that we must advocate a system which extracts detail from the rendered image of an object rather than its geometrical description. This is because the rendered image is the actual information which is presented to the user's visual system. It is therefore the most accurate indication of perceived detail. Looking at the geometrical description of an object does not give a reliable indication of what the user eventually sees because the geometry can be displayed differently depending upon a number of factors such as: the particular shading model being used, the effect of any light sources, the use of texture maps or environment (reflection) maps, the simulation time of day (TOD), the use of fog or haze effects, transparency, etc.

More ideologically, if we are to develop a system based upon our perception of a scene, we should not have to concern ourselves with *how* the scene was produced, but only with *what* it contains. To use Marr's (1982) terminology, we should be concerned with the *representation* of an image, not the *processes* underlying it.

3.1.2 Spatial Frequency Analysis of Images

We have resolved that we wish to calculate the spatial frequency content of a computer-generated image. Now we have to consider how this analysis may be performed. Essentially, we wish to find a suitable mapping to transform a 2D function of intensity values into a 2D function of spatial frequencies. The solution which we will present here can be broken down into three independent stages. These are listed briefly below and will be further expanded presently.

1. **Feature Extraction** : find all of the atomic visual 'features' in an image. This is perhaps best achieved using an **image segmentation** algorithm with a suitable perceptually-based feature extraction mechanism.

2. **Spatial Frequency Calculation** : calculate all of the relevant spatial frequencies in each feature. At this stage, these frequencies are *relative* because they are in terms of pixels only; with no immediate concept of angular size, i.e. the units of spatial frequency at this stage are cycles per pixel (c/pixel).

3. **Spatial Frequency Transformation** : scale the relative spatial frequency values into units of c/deg. This transformation can be performed once we know the **field of view** (FOV) of the display device.

3.1.3 Review of Feature Extraction Techniques

Before we proceed any further, it would be useful at this point to briefly review some previous attempts which have been made to extract visual features from an image, or to otherwise assess its perceptual content.

There has been a large body of work in the area of **image quality metrics** (see Jacobson, 1995, for a recent review). However this field is dedicated to the concise perceptual evaluation of display devices. As such, these metrics are not concerned with the quality of arbitrary images displayed on these devices; and hence are of little benefit to us here.

Similarly, some work has been done in order to provide a numerical measure of the subjective distance between two images (Rushmeier *et al.*, 1995; Daly, 1993; Mannos and Sakrison, 1974; Gervais *et al.*, 1984), i.e. how perceptually different two images appear. These techniques try to map two arbitrary images to a single scalar quantity which describes how closely the images are perceived as equal. Such methods are therefore not of direct relevance to our problem because they are relative—not absolute—measures, and they do not describe the full range of spatial detail present in an image.

Within the field of computer vision a number of image segmentation algorithms have been founded on perceptual models (see Reed and du Buf, 1993, for a review). Many of these have resorted to Gestalt rules of grouping (Katz, 1951) to resolve all of the perceptually disparate components (e.g. Khan and Giles, 1992). Other feature extraction techniques have included attempts to generate a number of **feature maps** to locate 'meaningful wholes' in an image based upon a number perceptual criteria (e.g. Soufi and Scrivener, 1992), or the use of localised frequency domain techniques to categorise object groupings (e.g. Reed and Wechsler, 1990). In general, all of these algorithms are concerned with the identification of discrete *objects* within a scene, and not with the quantification of absolute detail within the image as a whole. (For example, the extraction of all pixels which describe a hammer; not the finer textural details which can be perceived within that object.) Also, many of these approaches have only been applied to binary or greyscale images: comparatively little consideration has been given to the analysis of full-colour images.

(As an aside, it is also common in computer vision systems to have to deal with noisy, real-world images. A number of techniques have therefore been developed to compensate for, or smooth over, noisy components within an image (Gonzalez and Woods, 1992). However our application does not need to concern itself with noise removal because the source images are computer-generated renderings, not digitised samples, and are therefore noise-free.)

We have already seen that the human visual system is thought to be composed of a number of parallel channels, each sensitive to a particular size of spatial detail. These channels can be modeled using 2D **Gabor functions** (Stork and Wilson, 1990) which are simply the product of a 2D sinusoid and a 2D Gaussian. Some researchers have therefore tried convolving an image using these Gabor functions in an attempt to develop and validate computational models of human vision (e.g. Watson, 1987; Beck *et al.*, 1987). For any single analysis, this convolution will report the degree to which a particular level of detail exists in an image. It does not report the entire range of perceptually visible detail which the image contains. That is, these techniques focus on a different, more basic, problem to ours: i.e. to what degree one particular visual channel is stimulated by a certain amount of detail, whereas we are effectively concerned with *which* channel(s) are optimally stimulated by a certain amount of detail.

No discussion of computational vision would be complete without mentioning the work of the late David Marr. Marr (1982) produced a machine-computable model for early human vision using edge detection techniques which could function over a range of scales. However, although the concepts of feature detection are relevant, this technique is of limited use to us here for the same reason that was given for the Gabor convolutions. Effectively, the size of a stimulus is an input to these detectors, whereas we want a detector which returns the size of a perceived stimulus as its output.

In conclusion, it can be observed that the requirements of our application are subtly different to all of those mentioned above. As such, a novel approach must be formulated. Let us therefore address each of the three stages of spatial frequency calculation which we have just identified in the previous section.

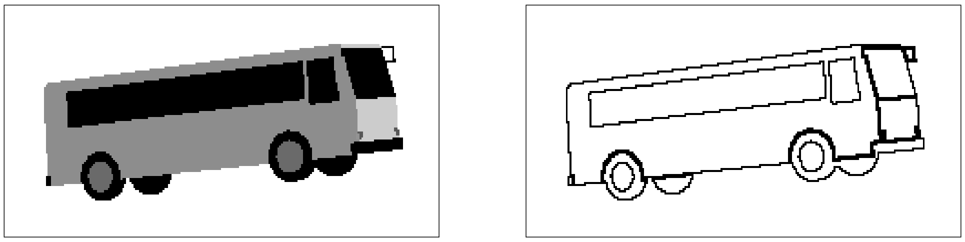
3.1.4 Extracting the Visual Features from an Image

The aim of this first stage is to extract all of the visually atomic 2D features within an image—the absolute elements of detail in the image. From a physiological standpoint, this would be the extent of a region which maximally stimulates a single neural channel in the vision system. Unfortunately, the mechanism which the human visual system uses to decide this delimitation is still unclear. We must therefore formulate our own, albeit simple, model for this process.

3.1.4.1 How to Define the Extent of a Feature

The mechanism which will be adopted here for the task of locating each visual feature in an image is based upon an image segmentation paradigm, i.e. a process which takes an image and segments it into a number of individual regions for independent analysis. This is done by taking a single pixel and then attempting to grow this pixel into a region; merging adjacent pixels with the region based upon a certain **segmentation criterion**. The crux of our dilemma therefore rests in the specification of this segmentation criterion.

The trivial case for defining the extent of a feature would be to only merge pixels which are exactly the same colour. This would be a valid definition if we were using a simple flat-shading polygon renderer (such as the Superscape VRT or REND386 graphics packages); or if we decided to use the flat-shading mode of our graphics renderer for time-efficiency reasons. Figure 3.1 illustrates this simple case of feature extraction.



(a) (b)

Figure 3.1: *(a) a flat-shaded image of a bus, and (b) a representation of the boundary of each visual feature in this image. In this case, a visual feature is simply a group of pixels with exactly the same colour.*

However, many contemporary graphics renderers offer far more sophisticated display algorithms than just flat-shading. For example, polygons can often be smooth-shaded, antialiased, and/or texture mapped. We should therefore be able to cope with these more realistic representations in addition to the simple case of flat-shaded primitives. Consequently, it is evident that we must relax the segmentation criterion slightly in order to include colours which are similar to the target colour, but not necessarily exactly the same. But to what extent can we relax this threshold?

In order to define this, we will utilise the body of research into **perceptually uniform colour spaces**, and **just noticeable differences** (JNDs). A perceptually uniform colour space is one in which the numerical distance between two colours (calculated with a suitable **colour difference formula**) is directly related to their perceptual distance. This distance is expressed as a number of JNDs. That is, we have a measure for the degree to which an observer perceives two colours as being distinct.

The reason for adopting this approach is to enable us to locate perceived edges in a colour image. The premise is that in a smoothly shaded region, one does not inherently notice the colour difference between adjacent pixels; but one perceives a gradual colour gradient over the region as a whole. Using JNDs, we can therefore decide whether two adjacent colours form part of a colour gradient (i.e. part of a feature) or are perceived as an edge between features.

3.1.4.2 In Search of a Colour Difference Formula

There is a bewildering array of colour difference formulae to chose from, each differing slightly from the next with regard to some attribute or parameterisation. However, before we converge upon any particular formula, it would be useful to take a cursory look at the background of these systems.

In 1931, the CIE (Commission Internationale de l'Éclairage) proposed the XYZ tristimulus coding scheme to describe colours in a device-independent manner. This provided an accurate and convenient means of specifying an absolute colour, but it did not consider the perceptual relationship between points in the colour space. In fact, the perceptual non-uniformity of the CIEXYZ colour space is about 80:1 (Poynton, 1993), i.e. the discrepancy between the perceptual distance and the numerical distance of two CIEXYZ colours can vary at points by up to about 8,000%.

It was not until 45 years later, in 1976, that the CIE introduced the CIELAB and CIELUV specifications which attempted to provide a more perceptually uniform colour space. These two systems managed to improve the perceptual non-uniformity of the CIEXYZ system to around 6:1. Since that time, a number of other systems have been proposed which try to improve upon the CIE's standard, e.g. the CMC formula (CMC, 1989), the BFD formula (Luo and Rigg, 1987), the FMC1 formula (MacAdam, 1985), etc.

The most relevant colour difference formula for our purposes is the CIELUV system. Most formulae were developed for use in the colourant industry as a product tolerance measure. These systems are therefore designed for viewing under reflected light conditions and are not directly applicable to self-luminous colours such as those generated by visual display units (VDUs). However, the CIELUV system was specifically devised for viewing additive sources such as colour monitors (Travis, 1991; Carter and Carter, 1983) and is therefore the best colour system for our purposes (Cowan and Ware, 1985).

Also, the CIELUV system is the most flexible because it is the only system which can incorporate size corrections (Carter, 1996). All colour difference equations are based around the CIEXYZ system which is only defined for colours that occupy 2 degrees of the user's field of view. But our colour perception degrades as the stimulus size decreases (Wyszecki and Fielder, 1971). Silverstein and Merrifield (1985) therefore produced scaling factors for the CIELUV colour space to model the effect of stimulus size on colour perception. We can therefore use their equations to better represent the perceived colour difference for very small stimuli, such as single or small groups of pixels.

3.1.4.3 Implementing the Feature Extraction Stage

Taking into consideration the above discussion, we can envisage a system which locates a visual feature in an image by recursively merging all adjacent pixels (or groups of pixels of the same colour) based upon the perceptual criterion which we have just introduced. That is, whether the group of pixels to be merged is within the prescribed JND threshold of the adjacent pixels already in the feature; using a size-corrected CIELUV colour difference equation (we will discuss the specification of this JND threshold later in Section 3.1.8.3).

Every pixel which is determined to be part of a feature could be copied to a **feature bitmap** in order to isolate the feature from the image. The feature bitmap does not need to hold full colour information for each pixel, only whether a pixel is in the current feature or not. The feature bitmap therefore need only contain binary information (e.g. a monochrome bitmap).

The contrast of the feature can be calculated by finding the ratio of the average luminance of each pixel in the feature, to the average luminance of the pixels immediately surrounding the feature (Peli, 1990).

In summary, a single iteration of the feature extraction stage produces the description of a single visual feature (e.g. a binary feature bitmap containing only those pixels which exist in the feature), and a value of contrast for that feature. This information is then passed onto the second stage for further processing. The above procedure is then repeated until all visual features have been found and processed. (N.B. Appendix B contains a number of examples of the feature extraction process for various images.)

3.1.5 Calculating the Relative Spatial Frequencies

We have now located a 2D visual feature in the image. The next stage is to calculate all of the appropriate relative spatial frequencies which define that feature. These frequencies will be relative because they are only in terms of pixels, i.e. they are a measure of the number of pixels a feature extends over. So the units of spatial frequency at this stage will be cycles per pixel (c/pixel).

3.1.5.1 Developing a Methodology

We have seen from Section 2.3.3.4 (Page 61) that in order to describe a 2D feature in terms of spatial frequency we must incorporate an orientation parameter. Spatial frequency is simply a measure of the size of a stimulus (to be correct, it is inversely proportional to size). Therefore, we effectively want to calculate the largest size of the feature—e.g. the longest contiguous line of pixels—at a number of orientations. Figure 3.2 attempts to illustrate this notion by presenting a feature along with three of its relative spatial frequencies.

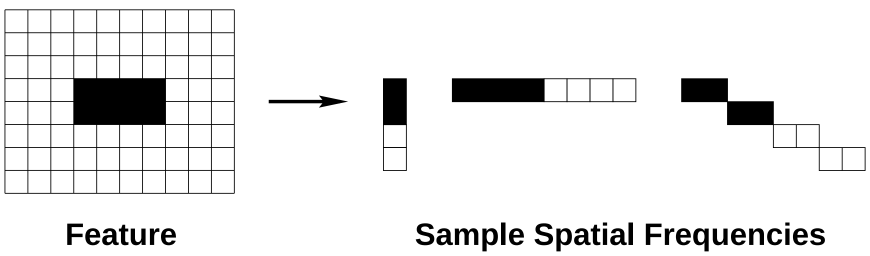


Figure 3.2: *A simple rectangular feature (left) and three of the feature's relative spatial frequencies at various orientations (right). The white pixels in the sample spatial frequencies are required in order to produce a complete contrast cycle in each case.*

If we take the case of the horizontal spatial frequency (0°) in Figure 3.2, then we can use this example to describe the method of calculating the c/pixel values for any feature. The longest horizontal line in the feature is 4 pixels in length, so the relative spatial frequency will be inversely proportional to 4 at that orientation. We can note that in terms of the bars in a square-wave contrast grating, our line of pixels is actually half a contrast cycle: a full cycle has a peak and a trough, i.e. two distinct features. We must therefore apply a scaling factor of 1/2 to our calculation to compensate for this fact. Putting all of this together, we can state that the horizontal relative spatial frequency of the feature in Figure 3.2(left) is: $1/2 \times 1/4 = 1/8$ c/pixels. That is, one complete contrast cycle over 8 pixels. This can be confirmed by looking at the middle spatial frequency in Figure 3.2(right).

From this example, we can develop a general relationship. If we know $l(\theta)$, the length of the longest contiguous line of pixels in a feature at orientation θ, then we can calculate the value of $RSF(\theta)$, the relative spatial frequency of the feature at orientation θ, as follows:

$$RSF(\theta) = \frac{1}{2l(\theta)}. \tag{3.1}$$

3.1.5.2 Implementing the Frequency Extraction Stage

Using Equation 3.1, we can find the relative spatial frequencies in a feature at any orientation. In order to resolve this however, we need to know how to evaluate $l(\theta)$: the length of the longest contiguous line of pixels in the feature at orientation θ. This was a trivial exercise in the example above because the feature was a simple rectangle; but how do we calculate $l(\theta)$ for more complex shaped features? This problem resolves to: how do we find the longest contiguous line of pixels, for a particular orientation, which can occur anywhere in the feature? The technique which the author used to solve this problem is described below.

Rather than attempt to formulate a mathematical solution based upon the geometry of the feature, a direct analysis of the feature bitmap was favoured: for any particular angle, a line of that orientation is notionally scanned through the feature bitmap. At each position, the largest number of contiguous (lit) pixels which exist on that line is recorded. The value of $l(\theta)$ is therefore the largest such result for every line which is passed through the feature. This process is repeated for any number of required orientations.

This method of direct analysis has the convenient benefit of implicitly handling all of the degenerate cases of feature shapes, e.g. concave features, and features with holes (sub-features). Figure 3.3 provides a depiction of this concept in three general cases.

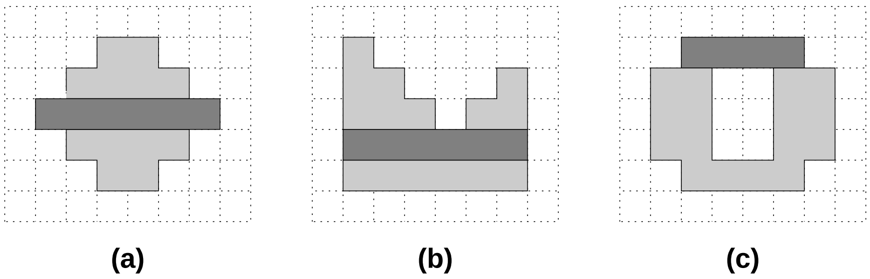


Figure 3.3: *An illustration of the maximum horizontal length of a number of features, showing (a) a convex feature, (b) a concave feature, and (c) a feature with a hole. The bold line of pixels represents the first occurrence of the longest contiguous line of pixels which will ultimately be used to calculate the spatial frequency at that orientation.*

3.1.6 Scaling the Relative Spatial Frequencies

The result of the previous two stages has been to calculate the relative spatial frequency for each feature in an image (at a number of orientations). These values are provided in units of cycles per pixel (c/pixel); however we want them to be available in units of cycles per degree (c/deg) in order to make any perceptual classifications. This naturally requires us to know the horizontal and vertical field of view which the display device occupies. In a head-mounted display system, this is a trivial matter because 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. This can be solved by referring to the tan rule for right-angled triangles, e.g.

$$FOV_{\text{horiz}} = 2 \times \tan^{-1}\left(\frac{width/2}{distance}\right), \tag{3.2}$$

$$FOV_{\text{vert}} = 2 \times \tan^{-1}\left(\frac{height/2}{distance}\right). \tag{3.3}$$

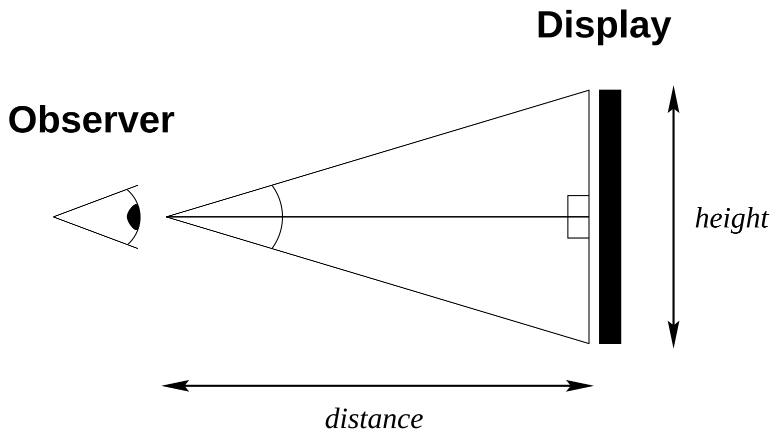


Figure 3.4: *Calculating the field of view of an arbitrary display device.*

From these transforms we can calculate the visual arc subtended by one pixel and hence convert the values of c/pixel into c/deg for all orientations. In the process of performing this, we wish to be able to support any arbitrary rectangular display FOV and any (not necessarily similar) rectangular display resolution. That is, the scaling factor in the vertical direction will not necessarily be the same as the scaling factor in the horizontal direction. In order to accommodate this, we can implement the process by: extracting the horizontal and vertical components of the frequency, scaling these components independently, and then recombining these to give the final result. These three steps are detailed below:

1. Extracting the horizontal and vertical components of the spatial frequency $RSF(\theta)$, at orientation θ, can be achieved by applying the trigonometrical formulae $\cos(\theta) = a/h$ and $\sin(\theta) = o/h$ for a right-angled triangle; where the hypotenuse of the triangle equals $RSF(\theta)$, i.e.

$$C_{\text{horiz}} = RSF(\theta)\cos(\theta), \tag{3.4}$$
$$C_{\text{vert}} = RSF(\theta)\sin(\theta). \tag{3.5}$$

2. In order to scale the value of c/pixel into c/deg, we need to know the resolution of the display device (in pixels) and its FOV (in degrees). Then, we can define the horizontal and vertical scaling factors as:

$$S_{\text{horiz}} = width/FOV_{\text{horiz}}, \tag{3.6}$$
$$S_{\text{vert}} = height/FOV_{\text{vert}}. \tag{3.7}$$

3. Finally, once both the vertical and horizontal components have been independently scaled, we can recombine the two components using Pythagoras' equation to give the resulting absolute spatial frequency, $SF(\theta)$, in units of c/deg:

$$SF(\theta) = \sqrt{(S_{\text{horiz}}C_{\text{horiz}})^2 + (S_{\text{vert}}C_{\text{vert}})^2}. \tag{3.8}$$

One final point which should be noted is the differential in perceived line length which is experienced with relation to orientation. E.g. a horizontal line of 10 pixels will appear shorter than a 45° line of 10 pixels. This can be easily explained by referring once again to Pythagoras' equation: a right-angled triangle with both perpendicular sides equal to 10, will have a hypotenuse of $\sqrt{10^2 + 10^2} = 14.142$, i.e. a 45° line of 10 pixels—which spans 10 pixels horizontally and 10 pixels vertically—will be perceived 1.4 times larger than a horizontal line of 10 pixels. We should therefore make an attempt to scale appropriately the relative spatial frequency value, $RSF(\theta)$, depending upon its orientation, before converting it into an absolute value, $SF(\theta)$. This can be achieved if we assume that a line of x pixels length (at any orientation) will always span either x pixels horizontally or x pixels vertically. This is the case for most line drawing algorithms, e.g. the standard Bresenham's algorithm (Kingslake, 1991). Therefore, we can calculate the value of $RSF'(\theta)$, which incorporates the appropriate compensatory scaling factor for any orientation, $\theta = [0..180^{\circ}]$, as:

$$RSF'(\theta) = \begin{cases} \sqrt{RSF(\theta)^2 + (RSF(\theta)\tan(\theta))^2}, & \text{when } 0^{\circ} \leq \theta \leq 45^{\circ} \\ & \text{or } 180^{\circ} \leq \theta \leq 135^{\circ} \\ \sqrt{RSF(\theta)^2 + (RSF(\theta)\tan(90^{\circ} - \theta))^2}, & \text{when } 45^{\circ} \leq \theta \leq 135^{\circ}. \end{cases}$$

3.1.7 What's Wrong with Fourier Analysis?

3.1.7.1 Introduction to Fourier Analysis

The technique of Fourier analysis can be used to decompose an image function into the set of harmonic intensity functions which sum to give the original image (Bracewell, 1965). This transformation is normally represented mathematically as $F(u,v) = \mathcal{F}\{f(x,y)\}$, where $f(x,y)$ represents the spatial domain of the original image and $F(u,v)$ represents the frequency domain of the Fourier transformed result.

The Fourier transform requires a continuous function to operate on; however, there exists a machine-computable method for calculating the Fourier transform of a discrete function which is called the Discrete Fourier Transform (DFT). The formula for the 2D DFT can be defined as follows:

$$F(u,v) \equiv \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}. \tag{3.9}$$

In practice however, computing this function directly is impractical. The most common method is to use the Fast Fourier Transform (FFT) which drastically reduces the complexity of the DFT calculation (Brigham, 1974).

3.1.7.2 Problems with Fourier Analysis

On first inspection, Fourier analysis sounds like a perfect solution to our problem: it takes a 2D source image and returns a frequency domain containing all of the relative spatial frequencies in that image. However, this is not actually what we require.

To illustrate this, Figure 3.5 presents the FFT of two simple images: a contrast grating and a square. In the first instance, the FFT of the contrast grating gives the expected results. There are three visible points. The centre point is the D.C. term and is not relevant to the analysis. The other two points represent a relative spatial frequency at orientations 0° and 180° which correlates with that of the contrast grating. (Note that the Fourier domain is symmetrical about the D.C. term.)

However, when we look at the FFT for the image of a single square, we are presented with a frequency domain which contains values across the entire spectrum of relative spatial frequencies (including ones which cannot possibly exist in the image). Part of the reason for this is that Fourier analysis only decomposes the sinusoidal intensity functions of an image, but the simple square image is effectively a square-wave intensity distribution and so it is rich in harmonics. Therefore, if we apply Fourier analysis to a square-wave intensity function (as might be produced by a flat-shaded feature in an image), then this will instantly produce spatial frequencies over the entire range of frequencies.

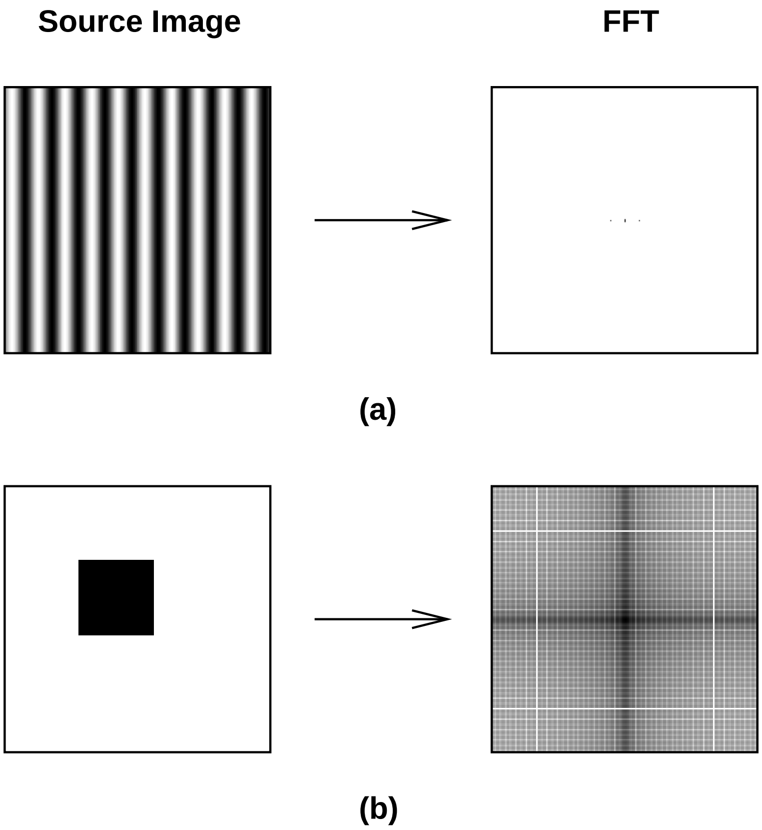


Figure 3.5: *The Fourier transform of two images: (a) an image of a contrast grating, and (b) an image of a square. The figures on the right represent the Fourier transform of the figures on the left, computed using the FFT algorithm.*

Specifically, we can conclude that Fourier analysis is not applicable to our application because of the following confounding factors:

1. Fourier analysis will only reveal harmonic spatial frequencies. Any square-wave variances will introduce substantial artifacts in the frequency domain. These cannot be removed from the frequency domain because, given any arbitrary point, we have no information to discern whether this was produced by a physical feature in the image or by the sine-wave approximation of a square-wave feature. That is, Fourier analysis will only (accurately) extract perfectly harmonic variances in intensity[1].

2. Fourier analysis will return values for all spatial frequencies at every point

[1]There exist other transforms, such as the Walsh Transform (Gonzalez and Woods, 1992), which can locate square-wave intensity components. However, these will then suffer from an inability to locate smooth intensity gradients, i.e. any such features will be recorded as a series of small square-waves instead of a more appropriate sine-wave coding.

in the image. So for example, if we take the situation with the image of the square: the 0° (horizontal) spatial frequency at the centre of the square will be relatively low, but the 45° frequency towards the top-left corner will be very high. That is, any 2D feature with a non-smooth boundary edge will always return frequency values across the entire spectrum.

The image segmentation algorithm circumvents these problems because it only extracts the *fundamental* spatial frequency (i.e. the lowest frequency) for every visual feature, at each desired orientation. The rationale behind only extracting the fundamental frequency is that if the lowest frequency is not detectable by the observer, then none of the higher frequencies will be detectable either—so we only need to record the fundamental frequency for a feature to decide whether it is visible under any specific viewing condition. This concept cannot be easily encoded into a Fourier process.

3.1.7.3 Spatial versus Frequency Domain Analysis

Given the above discussion, we can see that the image segmentation approach offers a number of advantages over the Fourier analysis method. These can be summarised as follows:

1. The image segmentation system can be applied to images that contain both flat and smooth shaded features (including antialiased and texture mapped features).
2. The corresponding frequency domain is noise-free with no high frequency artifacts. It is therefore much more amenable to accurate analysis.
3. The results are restricted to the fundamental frequency of each visual feature; instead of overloading us by reporting every spatial frequency in the image.
4. We have access to information about the physical position of a feature in the image (e.g. the position of a feature could be defined as the centre point of its bounding square). This is not available under the Fourier method.

The impact of these advantages can be easily ascertained by inspecting Figure 3.6. This presents a simple image with three squares of various sizes, and then displays the FFT of that image alongside the results of the image segmentation algorithm (plotted in Fourier space). As can be observed, the FFT result is highly noisy and it is unlikely that this could be of any use as an accurate measure of visual detail. On the other hand, the image segmentation process shows a concise and accurate result: it contains three discernible circles which represent the fundamental spatial frequencies of the three squares at each orientation.

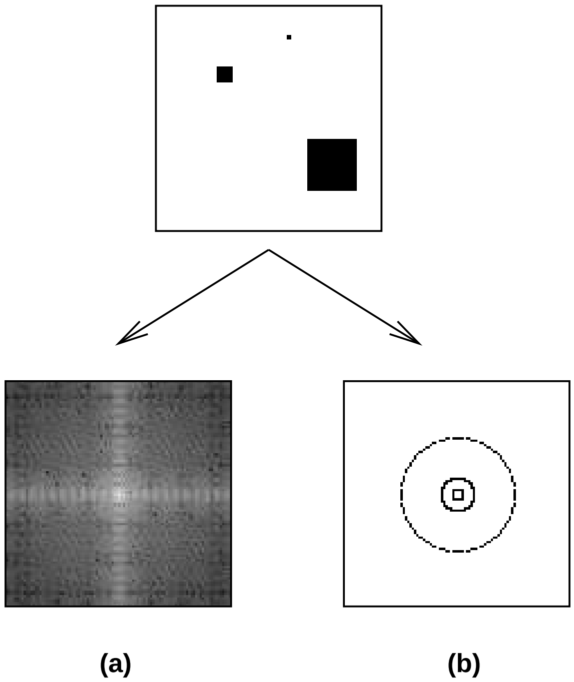


Figure 3.6: *A comparison of (a) the Fourier based, and (b) the image segmentation based approaches for locating the relative spatial frequencies within a sample image. (Both results are plotted in the Fourier domain.)*

It is certainly possible to investigate more sophisticated Fourier techniques, however it is the author's belief that this would be attempting to shoehorn a solution which patently does not fit in this situation. Although it is true that Fourier techniques have helped vision scientists to understand how the brain processes simple harmonic stimuli such as contrast gratings, recent theories of visual processing indicate that spatial detail is analysed on a local basis and that the visual system does not operate under Fourier principles (Daugman, 1984; Ware and Knight, 1995).

3.1.8 Discussion of the Measure for Perceived Detail

3.1.8.1 An Assessment of the Adopted Approach

We have presented a technique to extract all of the important spatial frequencies from a full-colour computer-generated image. This works by extracting features from an image, computing their relative size, and then scaling these into units of c/deg once the display FOV is known.

Of these three stages, the feature extraction process is the most subjective; with the other two stages being relatively well defined. Unfortunately, the field of visual perception can offer us little help to resolve the problem of defining a single 'feature' in a complex image. We have therefore devised a method which effectively locates the edges of features based upon whether there is a perceived colour differential between adjacent groups of pixels. This has some basis in the neural anatomy of the visual system (e.g. receptive fields are designed to located edges in the retinal image); however, it should be noted that there is no formal basis for such an approach[2].

As such, this model is certainly not proposed as a general model for how we perceive detail. However, it does perform well for the subset of computer-generated images, i.e. the model works best when there is no noise in the image and when surfaces exhibit smooth variations in colour. Both of these conditions are prevalent in computer-generated imagery, but less common in the real world.

One question which the discerning reader might be asking themself at this stage is why do we attempt to extract features based upon chromatic flux when we have already stated that we will use achromatic threshold data to model the visibility of a feature? The answer to this is that the purpose of the feature extraction process is to capture the *suprathreshold* perception of a scene. We can then apply the contrast sensitivity model to discover which of these features are below threshold (not visible) for any given viewing condition. If we segment the image based only upon the luminance information then we might not obtain an accurate perceptual segmentation for isoluminant colours. For example,

[2] As an interesting excursus, we can draw some qualitative comparisons between our approach and the Gestalt principles of organisation, e.g. we attempt to merge neighbouring parts of an image together (proximity), pixels are merged if their colour is sufficiently close to the rest of the feature (similarity), and features are solid 2D collections of pixels (closure).

consider a berry bush with green foliage and bright red berries. When we look at this scene in colour we are immediately aware of the berries as discrete entities (features). However, if we only look at the luminance information for this scene then we hardly notice the berries (if at all) because they have a similar intensity to the surrounding foliage. Mullen and Kingdom (1991) present a similar example and state that: '*any chromatic differences* [...] *would therefore be particularly useful for perceptual segregation*'.

3.1.8.2 An Evaluation of Colour Difference Formulae

Unfortunately, there is currently no accurate or principled measure for perceived colour difference, e.g. we have already noted that even the CIE's standard colour spaces (CIELUV and CIELAB) can exhibit a $\sim$6:1 error. In fact, Hardin (1988) suggests that it is not possible to find a single scheme to accurately represent all of the basic appearance attributes of surface colours.

The problem is that individuals' colour perception seem to vary quite widely. Wyszecki and Fielder (1971) report that agreement of results between any pair of observers is poor, and that even the ability of a single observer to match the same colours varies substantially over time. They state that averaging the data of several observers does not yield a representative result for an intermediate observer.

In addition to this subjective problem, there are a number of physical factors which can affect the appearance of colours on a computer graphics display (MacDonald *et al.*, 1990; Meyer and Greenberg, 1980). Some of these are beyond our simple control (such as the ambient light in the user's environment). However, some of the more accountable factors include:

1. **Stimulus Size** : colour differences vary in relation to the size of the two colours, but the CIE colour spaces are only defined for colours occupying 2 deg of visual arc. However, we have already noted that size correction formulae exist for the CIELUV colour difference equation (see Carter, 1989, for a review).

2. **Surrounding Colour(s)** : the brightness of a colour can appear different for different surrounding colours. This effect is know as **simultaneous contrast**. Phillips (1986) reports that this can induce perceived colour

difference variations in the order of 10–20%. However, when compared with the variability of a typical observer, this error does not appear to be significant (Phillips, 1986). More critically, this effect is of course the result of our human perceptual processing (the wavelength of light reflected by an object does not physically change due to juxtaposition with another object). This processing occurs at a later stage than the early visual detection stage that we are concerned with. We therefore do not need to compensate for this phenomenon in our application.

3. **Display Factors** : a number of monitor characteristics can affect the perceived colour of a stimulus. For example, a monitor's black level (or brightness) and its gamma value (Poynton, 1993). Care should therefore be taken to ensure that the display's brightness control is set so that dark elements are reproduced correctly, and that all colours are gamma corrected (see Reddy, 1996a).

It is therefore evident that any colour difference formulae which we employ will not give us a consistent and reliable estimate for a user's ability to discern between two colours. However these are the best measures that we currently have at our disposal.

3.1.8.3 Specifying the JND Threshold for Feature Extraction

We have left until last the question of how to specify the JND threshold for the feature extraction process. That is, what is the value of CIELUV colour difference (ΔE^*_{uv}) below which the user cannot perceive any change between two colours? This topic is not tackled to any extent by the colour perception literature; although a few references talk about CIELUV units in the order of 3–10 (Barten, 1990). We therefore decided to try and evaluate this threshold empirically.

An experiment was devised in which an observer tried to match two different colours so that they could perceive no difference. The ΔE^*_{uv} value at this point was noted in each case. Appendix A presents the results of this experiment. From these data we can observe a vast variability in the ΔE^*_{uv} results (ranging from $\sim$2 up to $\sim$30 CIELUV units).

Given the inherent problems of specifying colour perception which we have just described, the final decision of the threshold figure is quite arbitrary. We therefore decided to take the mean ΔE^*_{uv} value for all of the experiment trials. This evaluated to 10. I.e. within our measure for perceived detail, two pixels were considered equal if their size-corrected CIELUV colour difference fell below 10.

3.1.8.4 Example Spatial Frequency Analysis

To help illustrate the technique that we have just described, Figure 3.7 presents an example analysis with three different levels of detail for a model of a die. Next to each LOD is a graph of the relevant spatial frequencies in that image. The abscissa represents increasing spatial frequency (c/pixel), i.e. regions of high detail are coded to the right of the graph. The ordinate denotes the number of features in the image with a particular spatial frequency.

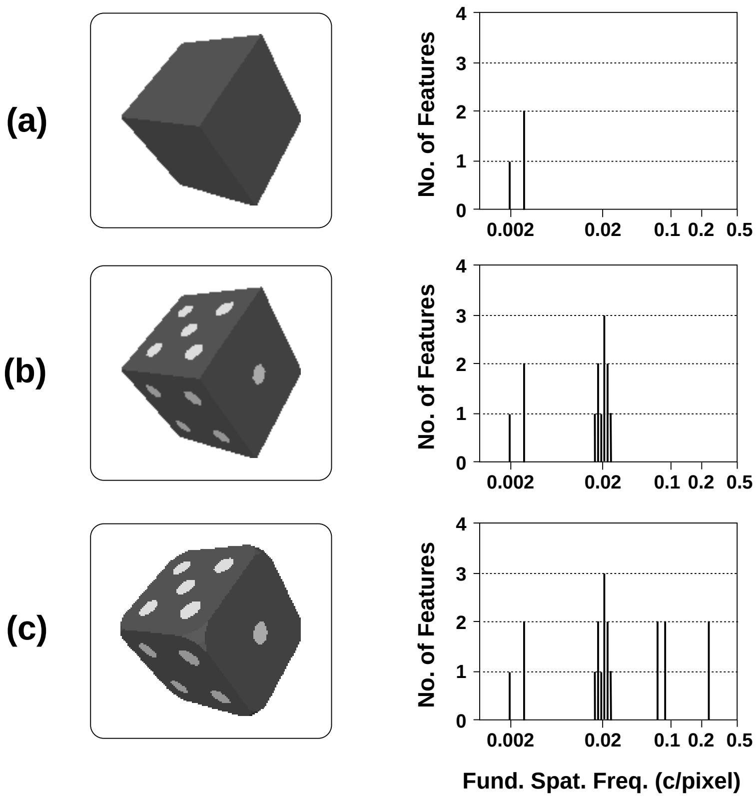


Figure 3.7: *An example spatial frequency analysis of three different LOD models. In each graph, the abscissa represents increasing spatial frequency (c/pixel).*

From these results we can note that there appear to be three major groupings of spatial frequency. In the case of the lowest LOD, Figure 3.7(a), we see that there are only three low spatial frequencies. These will obviously represent the three visible faces of the die; which are the only features at that level. In the medium LOD, Figure 3.7(b), we can see that we still have these three low spatial frequencies, but we now also have a batch of higher spatial frequencies. These will represent the spots on the die which have been introduced at that level. Finally, in the highest LOD, Figure 3.7(c), we can see the same trends as in the other two instances, but we also have some even higher spatial frequencies. These will represent the added detail induced by the curvature of the die at its edges.

From this analysis we can see that we have a high LOD which contains a number of high spatial frequencies, and that for each lower LOD we have fewer of these high spatial frequencies (less high detail). Therefore, if this die was presented to a user in such a situation where they could only perceive frequencies below 0.01 c/pixel, then we could select the lowest LOD model—Figure 3.7(c)—and the user would not be able to perceive any change.

3.2 Modeling Contrast Sensitivity

Now that we have formulated a metric to assess the perceptual content of an image in terms of spatial frequency, we require a means to predict how much detail a user can perceive in an image at various velocities and eccentricities.

We have seen from the previous chapter that the CSF defines the efficacy of the human visual system to resolve a stimulus based upon its size and contrast. However, these curves are produced by interpolating tabulated values obtained through empirical studies. In order to usefully incorporate these results into a computer graphics system we need to find a mathematical model to describe the shape of the human CSF.

Kelly (1975) embarked upon just such an activity. He developed a conceptual model to describe the spatial frequency characteristics of retinal receptive fields at high illuminance levels ($>$ 900 **trolands**), and showed that this can be used to model the sine-wave sensitivity of the visual system. Kelly's abstract model for contrast sensitivity is defined as:

$$F(\alpha) = \alpha^2 \exp(-\alpha), \tag{3.10}$$

where α represents spatial frequency (c/deg). (Recall that contrast sensitivity is defined as the reciprocal of threshold contrast, i.e. threshold contrast can be modeled by $1/F(\alpha)$.)

With suitable scaling factors, this general equation can be used to model the shape of the CSF under various viewing conditions. For our purposes, we wish to take into consideration the effects of velocity and eccentricity. We will therefore investigate how these variables can be incorporated into Equation 3.10.

3.2.1 Incorporating Velocity into the Model

The surface which is produced by mapping the CSF for a range of velocities is called the **spatiotemporal threshold surface**. This has been investigated by a number of vision researchers over the years. The most notable of these is D. H. Kelly, who was engaged in the study of the spatiotemporal surface for over 20 years. Our model will be based upon his findings. It is worth noting however that Burr and Ross (1982) conducted similar experiments to those of Kelly, and that their principal results correlate almost exactly with his.

Kelly (1979) made extensive studies of the spatiotemporal surface under conditions of stabilised vision[3]. From his data, he noted that the shape of the CSF remains essentially constant for all velocities above 0.1 deg/s, and only undergoes translation with increased velocity. He subsequently extended Equation 3.10 to model the spatiotemporal threshold surface (for velocities above 0.1 deg/s), by introducing two scale factors: k (the height of the CSF), and $\alpha_{\max}$ (the peak frequency of the CSF). His equation can be presented as follows:

$$G(\alpha, v) = kv\alpha^2 \exp(-2\alpha/\alpha_{\max}), \tag{3.11}$$

where,

$$k = 6.1 + 7.3|\log_{10}(v/3)|^3, \tag{3.12}$$

[3]A non-contact method of accurately stabilising the retinal image was used. This enabled measurements to be taken without the artifacts introduced by uncontrolled eye movements.

$$\alpha_{\max} = 45.9/(v+2). \tag{3.13}$$

Where v represents velocity measured in units of deg/s, and α represents spatial frequency in units of c/deg. Equations 3.11–3.13 can be combined into a single expression to give:

$$G(\alpha, v) = [6.1 + 7.3|\log_{10}(v/3)|^3] v\alpha^2 \exp[-2\alpha(v+2)/45.9]. \tag{3.14}$$

However, the author was unable to reproduce the empirical data which Kelly (and others) presents using this formula. Therefore, a modified version of Equation 3.14 was developed to more accurately model the available data. This was determined empirically with computer-aided techniques. The final solution was obtained by effectively altering the weighting of the k and $\alpha_{\max}$ components (with weightings of 41.0 and 2.75, respectively), and converting the base of the exponential term to base 10. The resulting equation can be represented as follows:

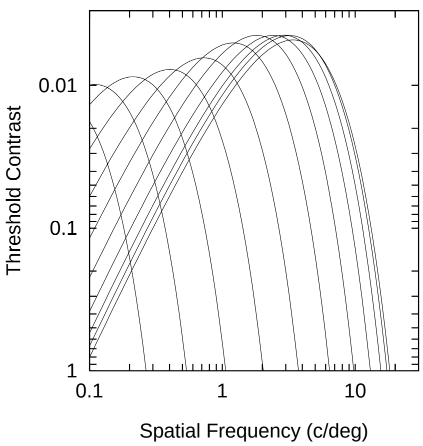
$$G(\alpha, v) = [250.1 + 299.3|\log_{10}(v/3)|^3] v\alpha^2 10^{-5.5\alpha(v+2)/45.9}. \tag{3.15}$$

This equation is plotted in Figure 3.8(a) for a number of velocities. From this we can observe that the effect of velocity on the CSF is to push the curve further towards the ordinate for higher velocities. This enforces what we have already stated back in Section 1.4.4.3: that we can see less high detail with increasing velocity. Figure 3.8(b) attempts to portray this relationship more clearly by plotting the highest visible spatial frequency for a range of velocities.

3.2.2 Incorporating Eccentricity into the Model

Contrast sensitivity declines with increasing eccentricity. However the shape of the spatiotemporal surface is consistent across the visual field (Virsu *et al.*, 1982; Koenderink *et al.*, 1978b; Kelly, 1984). This would lead us to believe that we can predict the contrast sensitivity for any region of the retina by simply scaling the foveal response with a factor based upon eccentricity.

Rovamo and Virsu (1979) confirmed this when they showed that visual acuity can be accurately predicted for any eccentricity by applying a constant scaling



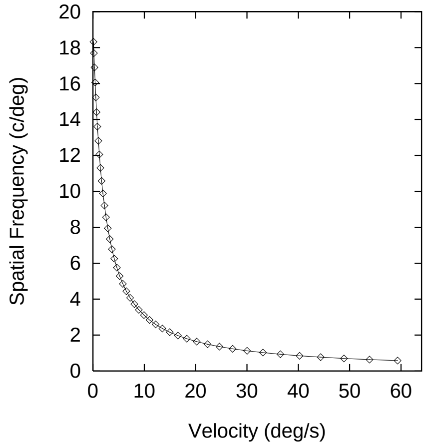


Figure 3.8: *(a) Contrast sensitivity functions for velocities of 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, and 128 deg/s (from right to left); calculated using Equation 3.15. (b) The highest visible spatial frequency (visual acuity) for a range of velocities, i.e. the upper point of intersection with the abscissa for each of the curves in (a). The data points represent calculated intersection points, with the curve illustrating the interpolated relationship.*

factor, referred to as the **cortical magnification factor** (M), first introduced by Daniel and Whitteridge (1961)[4]. Therefore, in order to incorporate eccentricity into our model of spatiotemporal contrast sensitivity, we simply need to apply this cortical magnification factor to Equation 3.15.

The eye's peripheral sensitivity is not circular symmetric (e.g. Regan and Beverley, 1983). For example, there are marked asymmetries between the **nasal** and **temporal** retina beyond 20 degrees (Sutter and Tran, 1991). Taking this into consideration, Rovamo and Virsu produced four equations to characterise M for each principal half-meridian of the retina. These are replicated below, and plotted in Figure 3.9(a).

$$\text{Nasal:} \quad M_N = M_0/(1 + 0.33E + 0.00007E^3), \quad 0 \leq E \leq 60 \text{ deg.} \tag{3.16}$$

$$\text{Superior:} \quad M_S = M_0/(1 + 0.42E + 0.00012E^3), \quad 0 \leq E \leq 45 \text{ deg.} \tag{3.17}$$

$$\text{Temporal:} \quad M_T = M_0/(1 + 0.29E + 0.000012E^3), \quad 0 \leq E \leq 80 \text{ deg.} \tag{3.18}$$

$$\text{Inferior:} \quad M_I = M_0/(1 + 0.42E + 0.000055E^3), \quad 0 \leq E \leq 60 \text{ deg.} \tag{3.19}$$

[4]Where M^2 is directly proportional to the density of receptive fields of retinal ganglion cells (Drasdo, 1977).

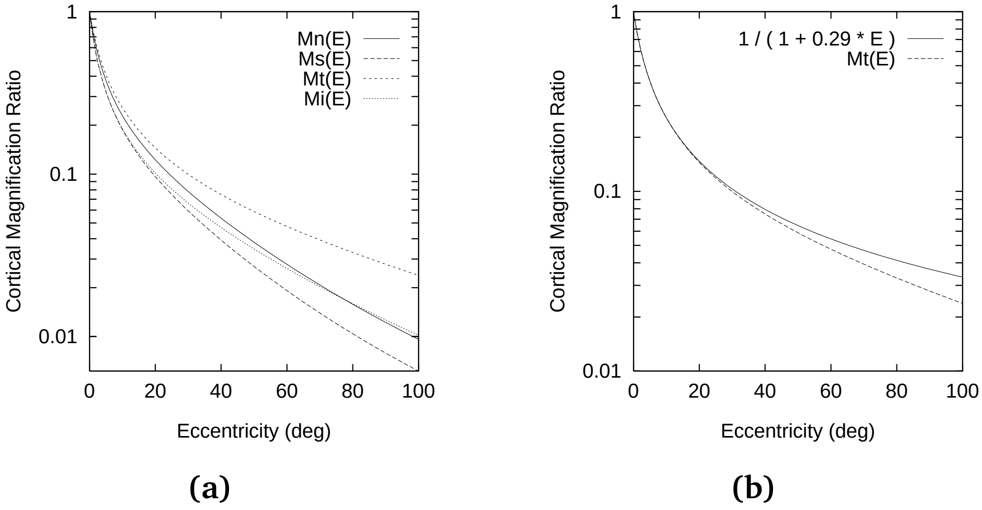


Figure 3.9: *(a) A graph of Equations 3.16–3.19, which define the cortical magnification factor, M, for each cardinal half-meridian of the retina. (b) A comparison of Equation 3.18 with the cubic term included (lower curve) and with the cubic term ignored (upper curve).*

Where M_0 is the value of magnification for the most central point in the fovea; which we can simply instantiate as $M_0 = 1$. An intelligent system would take into account the relevant region of the retina which is being considered and apply the appropriate value of M for that region. Alternatively, to simplify the relationship, we could exclusively use the most sensitive region's M, with the knowledge that the other regions will not exceed this sensitivity; i.e. Equation 3.18 (M_T).

It would be reasonable to ignore the cubic term in Equation 3.18. This only becomes significant at large eccentricities; and even when $E = 100$ deg, there would only be a $\sim$1% error. This simplification was adopted by Watson (1983), Kelly (1984), and Tyler (1985) in their respective models, among others. We can therefore define the cortical magnification factor for our purposes as:

$$M = M_0/(1 + 0.29E). \tag{3.20}$$

We can subsequently incorporate this equation into our model for contrast sensitivity as follows:

$$\begin{aligned} H(\alpha, v, E) &= G(\alpha, v) \times M, \\ &= G(\alpha, v)/(1 + 0.29E). \end{aligned} \tag{3.21}$$

3.2.3 Modeling Visual Acuity

We have seen from Section 2.3.4.1 (Page 63) that it is possible to describe the visual acuity of an observer in terms of spatial frequency, given their contrast sensitivity. We will therefore illustrate how this may be achieved for our model of contrast sensitivity.

In essence, we only wish to calculate the upper point where the CSF intersects the abscissa. Beyond this point, detail is not visible; irrespective of the level of contrast applied. Figure 3.8(b) illustrates this notion by plotting the velocity of a stimulus against the highest spatial frequency which is visible at that velocity (where $E = 0$ deg). From this figure we can observe that our visual acuity drops precipitously as a stimulus moves with greater velocity, up to about 10 deg/s, and then begins to level off asymptotically.

The point at which the CSF meets the abscissa is defined as a contrast sensitivity of one, i.e. $H(\alpha, v, E) = 1$. The solution to calculating the highest detail which a user can perceive for any velocity and eccentricity can therefore be written as:

$$H(\alpha, v, E) - 1 = 0. \tag{3.22}$$

We then need to solve this equation in terms of α in order to discover the spatial frequency at which $H(\alpha, v, E) = 1$, taking the highest root as the threshold size. An analytical solution to this problem would be overtly complicated; requiring the computation of Lambert's W function (Corless *et al.*, 1993), or equivalent, to resolve the combination $\alpha^2 \exp(\alpha)$ which arises. A more tractable solution can be found by using an iterative method such as an interval halving algorithm or a Newton-Raphson technique.

3.2.4 Discussion of the Visibility Model

Equation 3.21 provides us with a means of estimating the contrast sensitivity of a human observer to a stimulus of any size (α c/deg), moving at a particular velocity ($v >$ 0.1 deg/s), and located at any arbitrary eccentricity (E deg). We have also shown how we can describe a subject's visual acuity using the same parameters.

The velocity part of our model has been derived from the work of Kelly (1979).

His formula was produced analytically by attempting to find a general equation to model the results from various drifting contrast grating tests. These experiments were performed under stabilised vision in order to find the vision system's ideal spatiotemporal response. However, we do not view objects in the real world under stabilised conditions (i.e. our eyes exhibit small uncontrollable movements which can introduce motion artifacts). There is also evidence to suggest that, to a certain extent, our ability to perceive detail in moving objects is a skill which can be improved with practice (e.g. Sekuler and Blake, 1994; Murphy, 1978). However, this is probably due to an improved ability to fixate upon a moving target (thus reducing its angular velocity) rather than an adaption of our visual system proper. In addition, Kelly's (1979) work was concerned with responses to periodic sinusoidal gratings. We are of course primarily concerned with the visibility of aperiodic targets and so we may find that Kelly's threshold surface will underestimate the actual threshold surface for our application. It is therefore likely that our velocity model may not accurately model our pragmatic ability to perceive spatial detail under varying velocity. However, the model is as accurate as contemporary visual perception can allow.

It is worth noting that our references to the term velocity have so far implied translational velocity (the angular velocity of a stimulus across the retina). However there is also the possibility of rotational velocity (the angular velocity at which a 3D stimulus rotates on its axes). The field of visual perception contributes no significant knowledge to help us deal with this situation. This is understandable because even today vision scientists are still concerned with simple 1D or 2D harmonic images. However, presumedly, the visibility of a rotating stimulus could be determined if we can resolve its rotational velocity into a translational velocity across the retina.

Our contrast sensitivity model also accounts for the eccentricity of a target by applying the work of Rovamo and Virsu (1979). Their formulae for cortical magnification were based upon published data for the density of ganglion cell receptive fields across the retina. We know that visual acuity varies linearly with M across the retina, and that this is due to a measurable increase in the smallest receptive field size of cells towards the periphery, i.e. this constitutes a fundamental and physical limit to our visual resolution. We would therefore expect that the cortical magnification factor which we have incorporated into our model should provide an acceptable estimate of a user's peripheral sensitivity to spatial frequency.

One factor which is notably absent from our model is the ambient illumination level. We know that our ability to perceive detail varies with respect to the background illumination; with our resolving ability degrading in darker surrounds (Kelly, 1975). However, Kelly's (1979) model was formulated for high background illumination. This is therefore a worst-case model, and so implicitly handles the case of low background illumination. We therefore do not need to consider the effects of ambient illumination in our contrast sensitivity model.

3.3 Perceptually-Driven LOD Generation

With a mechanism to compute spatial frequencies in a computer-generated image, and a computational model for contrast sensitivity, we have now taken care of all the vision related problems. The only other factor which we have yet to address is how to design a polygon simplification algorithm which supports and complements a perceptually-based LOD system. This will be the subject of the current, and final, section in this chapter.

3.3.1 Formulating an LOD Generation Framework

Subsequent to our review of the polygon simplification literature in Section 2.2, we formulated three principal criteria for the use of polygon simplification techniques in a perceptually-based LOD system. To recapitulate, these were: 1) the effect of the simplification should be perceptually predictable, 2) the surface properties of the model should be taken into consideration, and 3) a topology simplifying algorithm should be advocated.

With regards to the last criterion, we do not want to restrict ourselves unnecessarily to any one specific simplification algorithm, because every algorithm offers different advantages and disadvantages. Indeed, as Luebke (1996) notes, it is unlikely that we will find a perfect all-encompassing simplification scheme. We therefore desire a general framework which addresses the first two criteria, but which can support a number of techniques for polygon simplification.

We are already aware that our perception of visual detail is based upon the relative size of a stimulus. Therefore, in order to simplify a model in a perceptually predictable manner, we must be able to guarantee that the degeneration is restricted to a certain scale. The GENIE system—reported by Kemeny (1993) and described in Section 2.2.2.6—achieves this by placing a 2D grid over a projection of the model and then reducing the representation so that only one polygon occupies any one grid cell. This effectively removes all detail below any prescribed scale. However, the technique is limited to two dimensions, whilst we require a general 3D technique.

By extrapolating the GENIE concept, we can suggest a framework for restricting the extent of a simplification process in three dimensions: i.e. instead of using a 2D grid on a projection of the model, we can use a 3D grid on the model itself. Thus we can segment a model into a number of **sub-volumes** and locally perform a simplification algorithm on only those polygons which exist within any particular sub-volume.

By altering the size of the sub-volumes we gain the ability to vary the scale of reduction; e.g. a fine grid mesh will remove only very small features, whereas a coarse grid will remove larger features. The overall effect of this mechanism is to ensure that no changes are made to the model above a certain threshold size. Therefore, if we choose a grid volume which projects to a size smaller than that of a single pixel for a particular viewing distance, then we gain the ability to automatically remove any detail which will not be displayed on the output device.

3.3.2 Describing an Implementation of the Framework

We will describe here how our sub-volume based framework for polygon simplification can be implemented. Figure 3.10 below presents an abstract top-level algorithm to illustrate the general structure and functionality of the framework; whilst the following sections will take each of the major steps in turn and discuss these in greater detail.

3.3.2.1 Geometric Normalisation (Step 1)

The initial normalisation process is an optional preprocessing stage which attempts to provide the geometric information in a standardised and consistent

Step 1:	`Normalise the geometrical data`
Step 2:	`Segment the model into sub-volumes`
Step 3:	`For each sub-volume in the model:`
Step 3.1:	`    Find all polygons entirely within the sub-volume`
Step 3.2:	`    Apply a simplification algorithm to only these polygons`
Step 3.3:	`EndFor`
Step 4:	`Optimise the geometrical data`

Figure 3.10: *Top-level algorithm describing the operation of the sub-volume based polygon simplification framework.*

form. We have noted in the previous chapter that most simplification algorithms only accept models in a certain format (e.g. triangular meshes with no coincident triangles). The purpose of this step is therefore to make our algorithm more accessible by attempting to convert arbitrary polygonal data into a format which most simplification algorithms can utilise. This includes actions such as: triangulating arbitrary polygons; removing multiply-defined vertices, edges, and polygons; removing coincident polygons; etc.

3.3.2.2 Sub-volume Segmentation (Step 2)

The simplest way to segment a model into a number of sub-volumes is to find the bounding box of the object and then choose a granularity with which to divide this volume into smaller rectangular volumes. Figure 3.11 illustrates this concept by presenting a model with an example subdivision grid overlayed. We should note however that the sub-volume need not be rectangular in shape, e.g. a spherical volume could easily be used instead.

3.3.2.3 Sub-volume Restriction (Step 3.1)

For each sub-volume, all polygons which are *entirely within* that volume are located and passed onto the next stage for simplification. This ensures that only polygons whose extents are definitely smaller than the threshold volume are considered for reduction (i.e. polygons which constitute below-threshold detail). This task could be implemented efficiently using an octree data structure to organise the 3D model.

One limitation of this approach is evident: any polygons which lie on the bound-

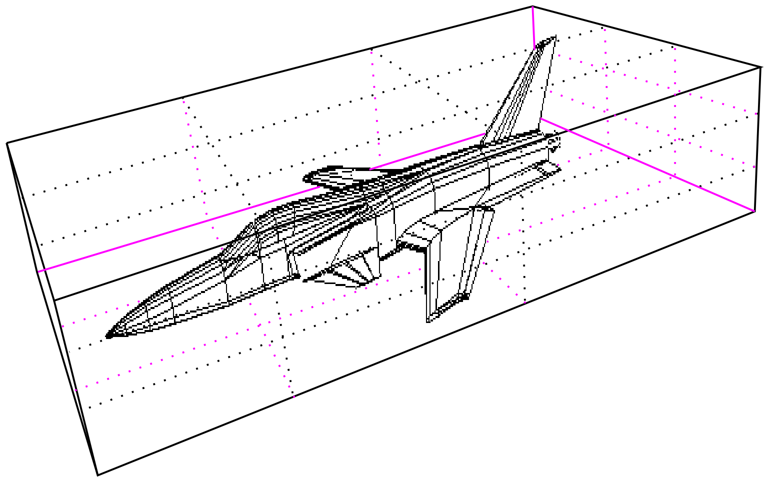


Figure 3.11: *An example model showing a sample sub-volume segmentation. The contents of each sub-volume is considered independently for simplification.*

ary of a sub-volume will never be considered for reduction (because they are not entirely within any sub-volume). The solution to this problem is to overlap successive sub-volumes. As a simple example, if the dimensions of the sub-volume are w, h, and d respectively, then possible positional offsets for a sub-volume can be given by: $(w \times l/2, h \times m/2, d \times n/2)$, where $l, m, n \in [0, 1, 2 \ldots]$.

3.3.2.4 Local Simplification (Step 3.1)

The simplification phase can be implemented using any of a number of polygon simplification algorithms, such as those presented in the previous chapter. For example, Reddy (1996b) presents an implementation of the sub-volume framework using an edge degeneration scheme. This proceeds by trying to collapse certain vertices so that they become co-linear with two of their neighbouring vertices. A reconstruction phase then attempts to rebuild the model in an optimal fashion, merging the resulting pairs of co-linear edges into a single edge.

The sub-volume restriction phase provides us with the assurance that the available polygon subset is entirely contained within the current sub-volume; and so the effects of the degeneration can be constrained to a predefined spatial boundary. However, it is likely that a polygon may contain a vertex which is common to a polygon not entirely within the current sub-volume (as illustrated in Figure 3.12). In this situation, if we alter the position of such a vertex, then we must consequently alter the representation of a polygon which is out with the sub-volume. This side-effect cannot be tolerated if we require that no visual changes are to be made to the model above the sub-division threshold.

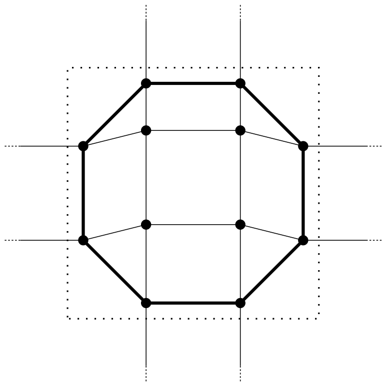


Figure 3.12: *A simple polygon mesh showing the extent of a sub-volume restriction (the dotted square). The bold demarcation illustrates the boundary edges within the sub-volume. Vertices on this boundary should not be considered for simplification.*

To enforce this requirement, a proviso must be added that no degeneration is attempted on a vertex which is shared by a polygon existing out with the current sub-volume (e.g. a vertex which lies on the boundary edge of the sub-volume).

3.3.2.5 Final Optimisation (Step 4)

The last step provides an optional postprocessing stage which attempts to optimise the final polygon mesh. For example, this stage could remove any unused vertices from the model's description, or merge any resultant adjacent, co-planar polygons with identical surface characteristics.

3.3.3 Dealing with Surface Properties

We have now addressed two of our three criteria, but we have yet to consider the effect of surface properties on the simplification. Most simplification algorithms attempt to repeatedly replace a set of adjacent polygons with a smaller set of adjacent polygons that retain the general characteristics of the original grouping. We therefore have to resolve how to define the surface characteristics of any new group of polygons (e.g. their colour, texture, etc.).

If we consider the function of the human visual system once more: our vision system has an amazing ability to integrate over features which are below threshold. For example, if we have a fine mesh of black and white squares and display this at a distance such that it is below the threshold of vision, then we can no longer resolve discrete black and white regions. Instead we perceive a single wash of colour which exhibits the average contrast of the two component colours (Livingstone, 1988). This phenomenon is used to great effect in the newspaper industry where halftoned patterns can be used to portray a wide range of greyscale levels with only a single black pigment on white paper.

To apply this principle to our problem, we can state that the colour of a new polygon, or group of polygons, should be based upon the average colour of all the previous component polygons, i.e. the RGB values for each polygon are averaged. This results in an averaging of luminance (achromatic) as well as colour (chromatic) information, due to the linear relation between them. However, the case is not quite that simple because the relative sizes of the different colour stimuli will affect the perceived result. For example, if the black squares in our mesh were slightly larger than the white squares, then the below-threshold wash would appear darker than when the squares were equally sized. Therefore, the colour of a new polygon grouping is found by an *area-weighted averaging* of the RGB colours for all of the polygons which previously occupied the same volume.

The principles for dealing with texture mapped polygons are exactly the same. In fact, we can think of a single texture mapped polygon as fulfilling the same visual function as a collection of tiny, individually coloured polygons. Stated this way, we can suggest that a good replacement for a texture mapped polygon would be to use an untextured polygon whose colour is found by averaging all of the pixels in the texture map image. This also has the desirable side-effect of replacing textured polygons with less complex primitives. Texture mapped polygons normally incur a greater computational and memory overhead than simple flat or smooth shaded polygons. Therefore, removing these primitives should improve the rate at which the model is rendered, hence adding another facet to the simplification algorithm.

3.3.4 Discussion of Perceptually-Driven LOD Generation

This section has presented a general framework for performing polygon simplification which could be implemented using a range of techniques. The framework enables us to predict the perceptual side-effects of a simplification by restricting the spatial extent of a degeneration.

We introduced one constraint on the simplification algorithm to ensure perceptual predictability (that vertices shared by polygons not entirely within a sub-volume should not be affected). We also suggested an enhancement to enable an algorithm to handle the degeneration of surface properties as well as geometric information (i.e. an area-weighted averaging of the previous surface properties over the extent of the local change).

The result of this development is that we can now accurately predict the screen size at which a model can be selected without introducing any visual artifacts. In many cases this may be an overestimate (e.g. if the largest change is well below the chosen sub-volume size), but the important point is that it should never be an underestimate. It is worth reminding the reader that we will not use this method (the projected size of a sub-volume) in order to define our LOD thresholds; we will of course be using our more general perceptual criteria, i.e. our measure for spatial frequency and our contrast sensitivity model. However, we will see in the next chapter that being able to guarantee that a reduction is restricted to a certain scale will greatly aid and simplify the implementation of a perceptually modulated LOD system.

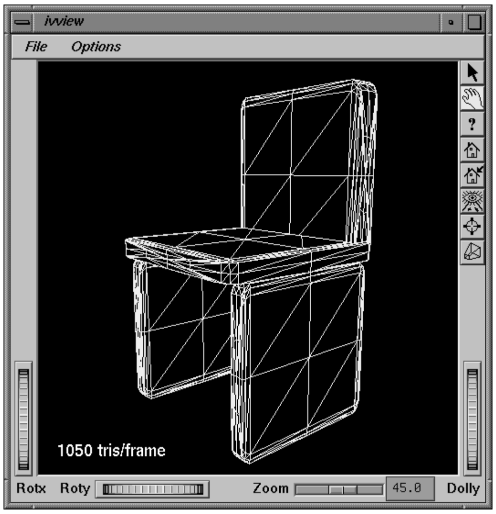
To illustrate the above concepts in practical terms, we present results from Reddy's (1996b) implementation of the framework. Figures 3.13(b)–(d) present the reduced versions of Figure 3.13(a), using sub-volume sizes of one sixty-fourth, one eighth, and all of the object's bounding volume, respectively (i.e. the widths of each sub-volume are one quarter, one half, and equal to the width of the bounding volume, respectively). This implies that the model in Figure 3.13(b) can be used when one sixty-fourth of the object's bounding volume projects to less than a single pixel on screen, and similarly for Figures 3.13(c) and (d).

Table 3.1 below reports the complexity of each model in terms of the number of polygons and vertices which it contains and the average frame rate which was achieved when it was rendered to the screen (using Open Inventor V2.0 on an SGI RealityStation)[5]. The absolute values of update rate are not important, because these will vary across platforms and graphics packages, but their relative magnitudes provide some indication towards the performance benefits which can be accrued when employing each degraded model.

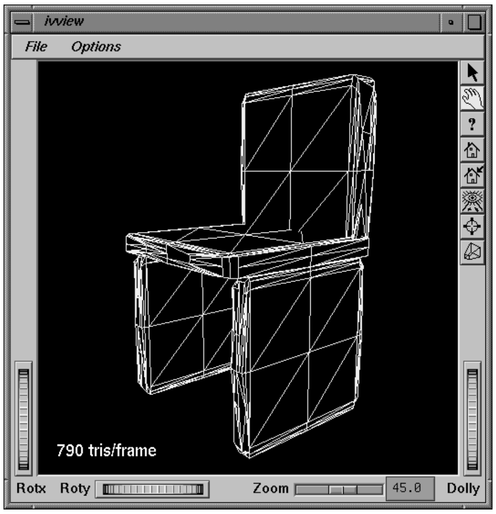
Model	**Polygons**	**Vertices**	**Update Rate**
Figure 3.13(a)	1050	524	5.9Hz
Figure 3.13(b)	747	496	8.1Hz
Figure 3.13(c)	300	302	17.2Hz
Figure 3.13(d)	186	242	30.2Hz

Table 3.1: *Geometrical and computational comparison of the degraded models presented in Figure 3.13.*

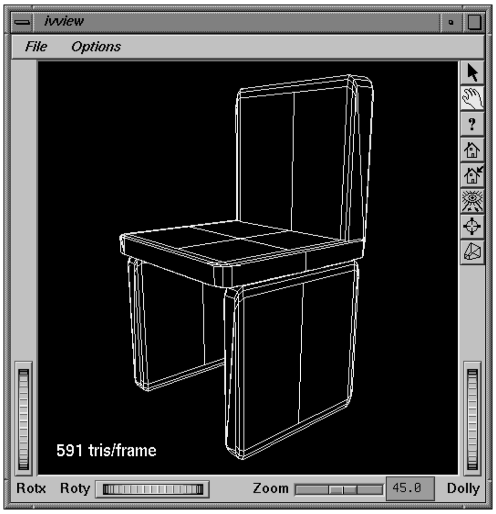
[5]Note that the table reports the number of (convex and planar) polygons in a model, whereas Figure 3.13 displays the number of triangles for each model (after undergoing an implicit triangulation process by the Open Inventor package).



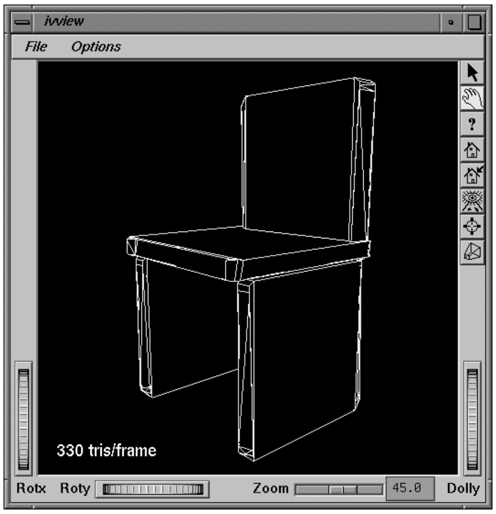
(a)



(b)



(c)



(d)

Figure 3.13: *Example reductions for a sample geometric model. The figures show: (a) the original model, (b) a reduced model using a sub-volume one sixty-fourth of the total volume, (c) a reduced model using a sub-volume which is one eighth of the total volume, and (d) a reduced model where the sub-volume equals the entire volume.*

3.4 Summary

As this chapter draws to a conclusion, we now possess tangible solutions to all of the outstanding problems which we identified as essential to the production of a perceptually-based LOD system.

In the first instance, we have developed a means of describing a computer-generated image in terms of its component spatial frequencies (c/deg). The approach which was adopted is based around an image segmentation process in which pixels are grouped into atomic visual features based upon JND determinations (using the CIELUV colour difference formula). These features are then analysed to extract their fundamental relative spatial frequencies (c/pixel); and then finally these values are transformed into units of c/deg based upon the FOV of the display device. During our discussion, we looked at Fourier methods and found these to be inapposite for our purposes: Fourier analysis is a valid technique for qualitative comparisons of image content or frequency filtering operations, but it has severe limitations as an accurate and concise measure for perceived detail.

Following from this, we produced a computational model for contrast sensitivity (Equation 3.21). Building principally from the work of Kelly (1979) and Rovamo and Virsu (1979), we developed a mathematical equation to predict the contrast sensitivity of a standard observer to a stimulus of any size, velocity, and eccentricity. We also illustrated how this model could be used to compute the visual acuity of the observer under the same conditions.

Finally, we formulated a general framework for polygon simplification which addresses the three criteria presented in Chapter 2. That is, the framework provides a mechanism to limit the scale of degeneration for any arbitrary simplification algorithm. This is achieved by segmenting the object's volume into a number of sub-volumes and independently applying a simplification algorithm to all of the polygons entirely within each sub-volume. In order to correctly degrade the surface property information, we apply an area-weighted averaging of the previous surface properties to any new triangulations.

With the above solutions in hand, we can now consider how these may be integrated to produce an efficient computer graphics system which modulates the detail of objects based upon their size, velocity, eccentricity, and perceptual content.

Chapter 4

Implementation

'Jim, we've absolutely got to have those communicators.
Without them we don't have the computers,
And without the computers we don't have a chance!'

(Dr. Leonard 'Bones' McKoy, *Star Trek: Miri*)

We now have sufficient pieces to our puzzle to embark upon a general model and subsequent implementation for a perceptually-based LOD system. We have a suitable measure for the degree of perceived spatial detail in a computer-generated image, and a model to predict the visibility of any part therein. Consequently we now have to consider how these can be combined to produce an efficient implementation which can be usefully integrated into a real-time computer graphics system.

4.1 A General Model for Perceptual LOD

Before proceeding further, it would be beneficial to briefly describe the general model that we are currently constrained to use due to the decisions which we have made thus far.

In the first instance, we are required to compute the spatial frequency content of each object off-line (because we have chosen a more visually accurate image based analysis rather than a geometry based technique). This will involve sampling the object from a number of different viewpoints in order to capture its 3D profile. By performing the spatial frequency analysis off-line we also alleviate much of the computational burden from the on-line scheduler (the code which performs the LOD modulation during the simulation), and hence simplify the time-critical component of our implementation. (For this reason many VE systems have included an off-line stage to perform any possible preprocessing, e.g. Airey *et al.*, 1990; Hitchner and McGreevy, 1993; Maciel and Shirley, 1995; Funkhouser and Séquin, 1993.)

Once on-line, the system must monitor the size on screen, angular velocity, and eccentricity of each object (in units of pixels, deg/s, and deg, respectively). It can then use this information to compute the highest resolvable spatial frequency under those conditions, using our model for contrast sensitivity. Also, given the spatial frequency data that were found from the off-line stage, we can estimate the instantaneous frequency content of each LOD. Finally, the system can then use all of this information to choose the optimal LOD for the object.

It is an obvious point, but one worth noting, that not all objects in a VE will need to be considered by the on-line scheduler. Only *degradable* objects need to be considered: i.e. those objects for which multiple levels of detail exists.

From the above, it is clear that our general model should be split into two fundamental stages: an off-line (preprocessing) stage, and an on-line (scheduling) stage. We can encapsulate the high level structure of each of these stages in the following two algorithms. The remainder of this chapter will be devoted to describing the implementation of these two stages in further detail.

Off-line :

```
produce a number of different LOD models for various objects
...
For each degradable object
   For each LOD of the object
      For a number of sample viewpoints around the LOD
         render the LOD
         extract all relevant spatial frequencies from image
         store perceptual attributes with model definition
      EndFor
   EndFor
EndFor
```

On-line :

```
perform any necessary initialisation
...
At each update of the VE
   For each degradable object
      find the object's size, velocity, and eccentricity
      evaluate the spatial frequency content of each LOD
      calculate the highest visible detail in the object
      choose the optimal LOD to use for the object
   EndFor
EndAt
```

4.2 Rationalising the Model

Thus far we have been concerned with the development of measures and tools which encapsulate contemporary models of visual perception as precisely as possible. However we do not need, or wish, to include this level of complexity in our system. Indeed, given the sheer complexity of our visual system, we could not hope to simulate all of the involved factors accurately and rapidly. Therefore, in order to entertain the notion of a real-time implementation, we are going to have to analyse the data requirements of our application more precisely, and rationalise our model where possible.

4.2.1 Optimising the Off-line Stage

With regards to the spatial frequency extraction process, we are obviously not concerned about every possible spatial frequency that exists within an object. Ideally, we would like to be able to find a single value to characterise the perceptual content of an object for our purposes.

In the first instance, we are not concerned with the absolute range of frequencies in a single object, but only with the *difference* between two successive levels of detail for the object. Take the example of two models of a calculator: one with buttons on the front face and one with no buttons. The spatial frequency of the calculator case is irrelevant because it appears in both models. It is the spatial frequency of the buttons which will dictate when the two models should be switched. Therefore, once we have collected the overall frequencies for all of the LOD models, we can go through and remove instances which are common to successive models, i.e. frequencies that have the same magnitude and location within the image. This will leave us with only the **unique spatial frequencies** for each model.

Performing this data compression can drastically reduce the amount of information that we must process. In addition to this, we can also streamline our model further by undertaking the following minor simplifications:

- **Ignore Contrast** : if we ignore the contrast of a feature (i.e. assume it to be 1.0) then we would effectively modulate detail based upon a user's visual acuity, rather than their contrast sensitivity. This would be a valid simplification, both from a perceptual and a computational standpoint, for the following reasons:

    1. As a result of performing our spatial frequency analysis off-line, we cannot account for any changes in the lighting of the model that may occur during the simulation. By ignoring contrast we therefore conservatively take the worst case scenario; thus helping to preserve our goal of being able to modulate detail without affecting the user's percept of the scene.

       (Also, by ignoring the contrast of a feature we can reduce the disk and memory overheads of our model.)

    2. As can be observed from Figure 3.8(a), our contrast sensitivity drops off precipitously beyond the optimal spatial frequency, e.g. taking the curve for 1 deg/s, if a feature dropped from full contrast (1.0) by an order of magnitude to 0.1, then the threshold spatial frequency would only change by a few units, from $\sim$13 c/deg to $\sim$10 c/deg. We would therefore be losing a relatively small degree of accuracy if we were to favour visual acuity over contrast sensitivity.

- **Ignore Orientation** : the orientation of a spatial frequency is of little importance when considering static targets[1] because the orientation of

[1]Except perhaps to compensate for the aspect ratio of the display.

a feature will not improve its visibility. Orientation only becomes relevant when considering moving targets because the visibility of a moving spatial frequency is dependent upon its alignment with the direction of motion.

For example, consider a long, thin, vertical feature. Its horizontal spatial frequency (derived from its width) will be very high, but its vertical spatial frequency (derived from its height) will be very low. If this feature moves horizontally, then at a certain velocity it will become invisible to the naked eye. If this feature then moves at the same velocity but in a vertical direction, it will remain visible because the vertical spatial frequency is far lower than its horizontal counterpart. This effect can be observed in the natural world: e.g. when you look fixedly out of the window of a fast moving train towards a meshed fence of horizontal and vertical wires, then you cannot see the vertical wires of the fence, only the horizontal wires which are aligned with your direction of motion.

Therefore, incorporating orientation into our model would only prove beneficial under circumstances where certain, significantly elongated features are travelling at high velocity and are perpendicularly aligned to the direction of movement. Even if this case is met, there may be other features in the object which do not meet these criteria and hence are visible, thus the level of detail of the object as a whole cannot be altered. We therefore propose that the computational effort required to account for feature orientation and direction would not be justified. Accordingly, we will ignore the orientation of spatial frequencies in our implementation. That is, we take the worst case scenario for each feature in an image, i.e. the lowest frequency that exists at any orientation in a feature.

4.2.2 Optimising the On-line Stage

Most of the data reduction that can be performed on our model is relevant to the off-line stage. As a result of this optimisation, we automatically optimise the on-line stage because the complexity of the data it must handle is reduced. There is therefore comparatively little rationalisation required for the implementation of the on-line stage. However, there is one algorithmic simplification that we will regard as reasonable to impose:

- **Ignore Rotational Velocity** : we have already noted that there are two distinct types of velocity: translational velocity and rotational velocity (see Section 3.2.4, page 94); where rotational velocity is the rate at which an object rotates about its axes. Modulating spatial detail using rotational velocity would be an involved process and one which does not fit easily into an object-based selection paradigm such as LOD. Also we have the problem that very little perceptual data have been reported to help us correctly account for the visibility of a rotating object.

  For example, when an object is rotating, then the translational component of any feature's velocity will vary for different phases of the rotation (e.g. the translational velocity component of a point on a rotating sphere varies sinusoidally as the sphere turns). Also, features further away from the point or axis of rotation will travel faster over the display (e.g. if the radius of the rotating sphere was increased, then the point would have to travel over a larger area in the same time period). We can therefore see that different features in an object could be travelling at different velocities across the retina. We could endeavour to find the worst case scenario once again and simply use this to model the effects of rotational velocity. However the worst case would be represented by the translational velocity of a feature located at the point/axis of rotation, which will be 0 deg/s.

  To be accurate, we would require a system that could manage detail at a feature or polygon level, rather than an object level. Such a system would require the tracking of each individual feature within each level of detail of each object; projecting the relevant 3D coordinate for each feature into screen space and calculating velocity based upon the feature's previous such value. It is extremely likely that such a technique would be computationally prohibitive.

  For all of these reasons we feel that a general model for seamlessly modulating an object's spatial detail based upon its rotational velocity cannot be achieved using the standard notion of LOD. We will therefore not attempt to incorporate this into our current implementation.

4.3 Implementation of the Off-line Stage

We have now produced a general model for perceptually modulated LOD, and rationalised this model for a real-time computer graphics system. We can now consider how this model may be implemented. Accordingly, we will here discuss the various issues that are associated with the implementation of our off-line stage. That is, the generation of a range of LODs for different objects as well as the sampling and processing of each LOD's (relative) spatial frequency profile. The descriptions in this section should provide the reader with sufficient insight into the exact operation of the off-line stage, and how it can be implemented.

4.3.1 Generating each Object's LOD

Naturally, in order to modulate the detail of an object, we need to be able to create different representations of that object at various complexities. Section 2.2 provided a review of numerous polygon simplification techniques which can be used to produce these levels of detail from an original object. Any of these could be used for the task of generating different LODs for objects. However, this thesis makes the recommendation that some framework should be enforced to restrict the spatial extent of a reduction. One such scheme was developed in Section 3.3.

The reason for this recommendation is that we need to be able to create a collection of objects that contain different ranges of unique spatial frequencies. If all levels of detail for an object contain unique spatial frequencies of the same magnitude, then there will be no situation when one model can be favoured over another based upon their perceptual content. Essentially, we require a collection of LOD models with a range of *visual* complexities, in addition to a range of *geometric* complexities.

4.3.2 Generating each LOD's Spatial Frequency Profile

The major component of the off-line stage (as far as this thesis is concerned) is the computation of each object's perceptual content: that is, the relevant spatial frequencies which are component in each LOD model. These data are then attached to the object's description (along with its other attributes such as geometry, materials, position, etc.) for later use by the scheduler to select the best LOD in any situation.

4.3.2.1 Gathering the Spatial Frequency Data

Before we can investigate the spatial frequency information within the rendered image of a model we must think about how we are going to present the model. There are a number of display considerations that we must address:

1. **Model Size** : how large should we display the object on the screen? If we display an object too small then we may lose some of the small detail differences between levels of detail. The obvious answer is to render the object as large as possible on the display device.

   More precisely, we wish to position the object at a distance from the viewpoint so that the object's **smallest bounding sphere**[2] projects to the largest possible screen area, with no part of the sphere projecting to a point out with the viewport. This placement will ensure that all of the object will be displayed on the viewport no matter what its orientation.

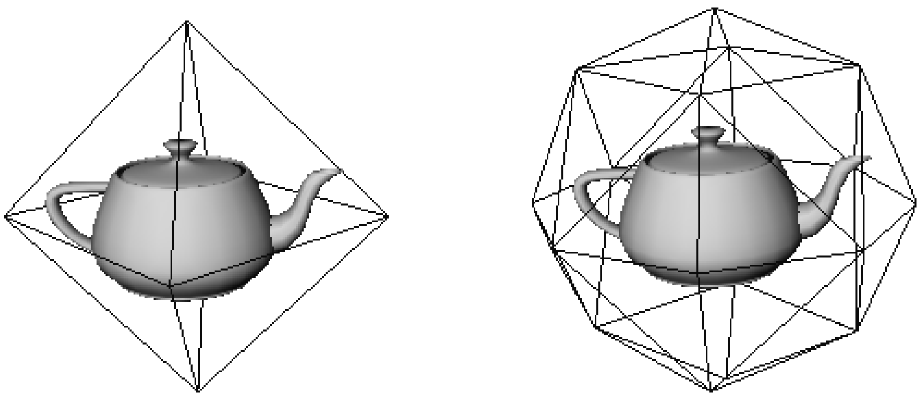
   A related issue here is also how to specify the size of an object (because we concluded in Section 2.1.7, Page 38, that size is a more resilient measure than distance for modulating detail). This is an arbitrary decision because the value will only be used in a relative sense, and so an absolute measure is not required. We will therefore favour the definition of object size used by Roehl (1995), because this offers a lightweight and orientation invariant measure. That is, the size of an object is represented by the length (in pixels) of the projected radius of an object's smallest bounding sphere. We record the size of an object when it is sampled because this information will be required by the on-line stage.

2. **Sample Viewpoints** : we have already noted that because our spatial frequency analysis is based upon a single 2D snapshot of a model at one particular orientation then we must endeavour to take multiple snapshots around the model in order to capture its 3D profile. But which viewpoints

[2] We define a 'bounding sphere' for our purposes as any sphere which completely encloses an object, and whose centre is coincident with the centre of that object. We will augment this term with the adjective 'smallest' when we wish to imply the singular such sphere which encompasses an object the most tightly.

should we choose to sample the model from, and how many viewpoints should we use?

Given the optimal distance between the object and the viewpoint that we have just resolved, the set of all possible viewpoints around that object will lie on the bounding sphere whose radius is equal to this distance (the so called **view sphere**). However, there is an infinite number of points on a sphere, and thus an infinite number of potential viewpoints. We therefore attempt to discretise the space of possible views around an object by sampling a subset of points on its bounding sphere. This can be achieved by using an appropriate **spherical tessellation** routine to produce a uniform and consistent distribution of viewpoints (e.g. Borgefors, 1992; Hwang and Yang, 1993). Figure 4.1 illustrates this concept by showing two possible samplings for an example object.



(a) **(b)**

Figure 4.1: *Two examples of the space of possible sample viewpoints around an object. In each case, the viewpoint locations are denoted by the vertices of the tessellated bounding sphere. All viewpoints are assumed to be oriented towards the centre of the sphere. (a) illustrates the simplest case of 6 sample viewpoints, while (b) shows a more complex example with 18 viewpoints.*

It is interesting to note that Maciel and Shirley (1995) used a similar method in their LOD system. They sampled the bounding hemisphere of an object off-line, calculating images at each sample point in order to choose the best model to display at any orientation. (Presumedly, Maciel and Shirley used a bounding hemisphere—rather than a complete sphere—because they were concerned only with objects that were positioned on a ground plane, and so the underside of these objects would never be visible.)

The question that we are left with is therefore: how many viewpoints should we sample around an object? This is difficult to specify optimally beforehand because the number will depend upon the visual complexity of the object in question (which it is the purpose of our analysis to eventually ascertain). Also, we will be constrained to only certain numbers of sample viewpoints, dependent upon the particular tessellation scheme which is used to triangulate the bounding sphere.

The simplest such tessellation would produce 6 sampled viewpoints, giving samples for the top, bottom, and four sides of the object, as in Figure 4.1(a). This is a crude sampling, but may be sufficient for some simple objects such as cubes, cones, spheres, etc. However, for more complex objects a higher sampling resolution will be required. In general, between 18 and 128 sample viewpoints provide the most effective results.

Choosing a larger number of samples will not necessary affect the performance of the system during a simulation; however it will substantially increase the time taken to generate the object's spatial frequency profile, and it will also increase the storage overheads for the object.

3. **Lighting Conditions** : the degree of lighting which is used can affect the visibility of features in an object. For example, taking the extreme case where there is no lighting at all, then no features will be visible in an object. We should therefore attempt to sample the object under lighting conditions similar to those that will be experienced during the simulation, or under the brightest lighting conditions if these will vary on-line.

   It is important to note that this need not be done to an accurate degree. Because we ignore the contrast of features, it is only necessary that a feature be visible; even if only slightly. Therefore, we find that as long as there is ample ambient lighting then this should be sufficient for our purposes.

4.3.2.2 Processing the Spatial Frequency Data

Having collected the spatial frequency information (c/pixel) for a number of 2D snapshots of the object around its view sphere, we then filter these data to produce a more manageable description. This will involve applying certain operations to each individual snapshots in isolation, and also applying operations based upon comparative judgements between snapshots of more than one LOD from the same viewpoint. We shall refer to these as **local filtering** and **global filtering** operations respectively. Taking the first of these, we can describe the processes involved in the local filtering phase as:

Local Filtering Operations :

1. **Contrast Filtering** : we have decided to ignore the effects of contrast; effectively assuming a contrast of 1.0 for all features. This simply means that we disregard all contrast values that were found by the spatial frequency analysis (or more efficiently, we instruct the analysis program to not spend time computing the contrast values in the first place).

2. **Orientation Filtering** : we have also decided to ignore the effects of orientation in our model. We should therefore record only the lowest spatial frequency for all sampled orientations in each feature. This means that we use the longest part of a feature to represent its size and so we never underestimate the feature's perceived size.

The result of applying these local filtering operations is to produce a spatial frequency profile which contains a single value for every feature in each 2D snapshot. This value provides a measure for the threshold frequency of that feature which is both orientation invariant and contrast invariant. However, we can improve upon this data compression and reduce the spatial frequency profile to a single value for an entire 2D snapshot by applying the following global filtering operations.

Global Filtering Operations :

1. **Locate Unique Spatial Frequencies** : we are only concerned with the spatial frequencies that differ between successive levels of detail. We can therefore remove all frequencies that exist with the same magnitude and position in successive LODs at each viewpoint[3].

   Once this has been done, we can then discard the position information from the remaining spatial frequencies as we no longer require these data.

[3]The efficacy of this process is aided if we use a polygon simplification technique that incorporates a mechanism to limit the spatial extent of a degeneration. The reason being that all polygons above a certain size will never be modified and so the resulting spatial frequencies that they contribute towards are normally unaffected between successive models.

2. **<u>Locate the Threshold Spatial Frequency</u>** : our ultimate goal is to find the single value for spatial frequency which defines the point at which the next LOD model should be chosen. We can specify this as follows: given two successive LOD models, l and $l+1$ (where $l+1$ is more detailed than l), we wish to find the lowest unique spatial frequency in model $l+1$ which is greater than the highest unique frequency in model l (if one exists).

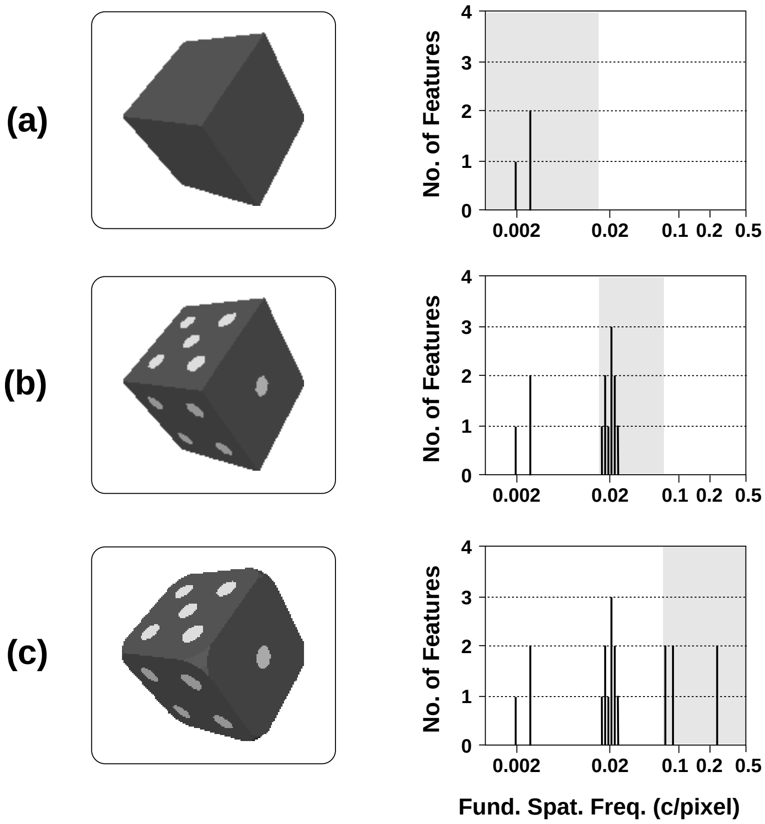


Figure 4.2: *An example illustration of the transition spatial frequencies for the die example we used previously in Figure 3.7. The shaded regions on the graphs show the frequencies over which the particular model can be used without visual defect. The value of spatial frequency which marks the boundary between two shaded regions represents the transition frequency for those two models.*

We shall refer to this value as the **transition spatial frequency** ($\alpha_{\rm trans}$) between models l and $l+1$. When we decide that the user is able to perceive spatial frequencies of this magnitude, then we must switch from the lower detail model (l) to the higher detail model ($l+1$). (N.B. if we have n levels of detail for an object, then we will always have $n-1$ transition spatial frequencies for each viewpoint.)

In simple terms, this procedure tries to keep the object at the lowest possible LOD. A more complex model is only chosen when the detailed features that it adds become potentially visible to the user.

Figure 4.2 illustrates the result of all these filtering operations on the die example we used in the previous chapter. We can see here that the transition spatial frequencies are determined by the lowest unique frequencies between two successive LODs, e.g. the value of α_{trans} between the lowest LOD, Figure 4.2(a), and the middle LOD, Figure 4.2(b), is around 0.015 c/pixel.

4.3.3 Focussing the Off-line Stage

Figure 4.3 attempts to encapsulate all of the various phases that contribute towards the off-line stage of our model. The overall result is to find a single value of spatial frequency (c/pixel)—for each sampled viewpoint—which defines the point when the next level of detail should be selected. Each of the five phases illustrated in Figure 4.3 are described below:

(a) **LOD Generation** : two different levels of detail are produced from the original die model. The models are produced so that the first model (left) only has very small features removed (the rounded edges), whilst the second model (right) also has larger features removed (the spots).

(b) **Sample Viewpoint Rendering** : both LODs are rendered from a number of viewpoints around the object's view sphere. We also record the projected size of the object at this stage so that we know the object size to which all the spatial frequency data pertain. The example here shows only 3 viewpoints for reasons of clarity. Note that these 3 viewpoints must correspond for all LODs, e.g. the first snapshot of the left-hand model was generated using the same viewpoint as the first snapshot of the right-hand model.

(c) **Spatial Frequency Analysis** : each 2D snapshot image is analysed to assess its spatial frequency content using the image segmentation algorithm that we developed in Section 3.1. Once this has completed, we will have segmented each image into a number of visual features, and for each feature we will know its position (x,y), spatial frequency (α), contrast (c), and orientation (θ).

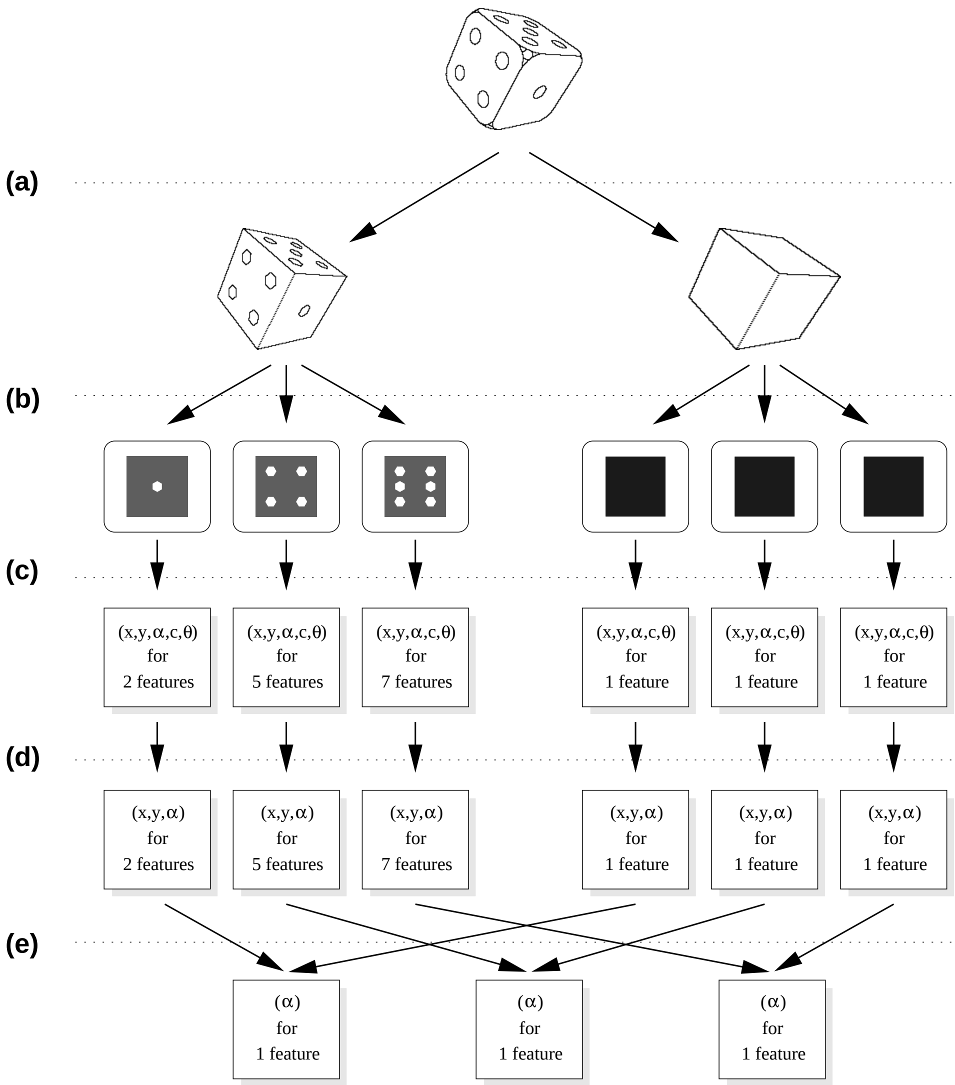


Figure 4.3: *An example overview of the off-line stage, showing the generation and spatial frequency analysis of two levels of detail of an original model. Phases (a)–(e) are described in the main text.*

(d) **Local Filtering of Data** : for each spatial frequency profile we remove the contrast data and find the lowest orientation value for each feature. This leaves us with only the position and spatial frequency attributes for each feature.

(e) **Global Filtering of Data** : we compare the data for both levels of detail at equal viewpoints. In each case we remove all frequencies which are common to both models and then discard the position information. Finally we find the threshold spatial frequency which defines when we should switch from one model to the other. This produces a single value of spatial frequency for each sampled viewpoint.

4.4 Implementation of the On-line Stage

Having discussed the details of the off-line stage, this next section will deal with the various implementation issues which are associated with the on-line stage. This will principally involve describing the task of selecting the optimal LOD for each degradable object, i.e. how to assess the perceptual content of each model on-line, how to evaluate the visibility of the object, and finally how to use this information to select the most appropriate LOD for an object.

4.4.1 Initialisation

The initialisation phase simply involves loading the results from the off-line stage and accounting for the specific viewing conditions, i.e.

- Load all of the spatial frequency information for each viewpoint of each LOD. (These values are currently in units of c/pixel.)
- Calculate the field of view of the display. This will generally be known for a head-mounted display, and can be calculated for a desktop display using Equations 3.2 and 3.3 (Page 77).
- Scale all of the relative spatial frequency values into units of c/deg based upon the specific FOV. This procedure is described in Section 3.1.6.

4.4.2 Monitoring Object Attributes

Our model of level of detail allows us to select an object's representation based upon its size on the display device, its angular velocity, and the degree to which it exists in the peripheral field. Our first task must therefore be to calculate these attributes of an object.

Object Size (σ pixels) : we have already stated on Page 112 that we will use the length of the projected radius of an object's smallest bounding sphere as a measure of object size. This can be calculated by finding the projected (screen) coordinate of the object's centre, and the projected coordinate of a suitable point on the object's smallest bounding sphere. We then simply calculate the Euclidean distance between these two points.

It should be noted that not just any point on the bounding sphere can be used to calculate the radius. We must choose a point which projects onto the circumference of the sphere's 2D projection, i.e. a point such that the 3D radius is perpendicular to the 3D vector connecting the viewpoint and the centre of the object.

Object Velocity (v deg/s) : we are concerned with the velocity of an object across the user's retina. We therefore need to find the rate at which the object moves across the display device (pixels/s), and then scale this into units of deg/s using the FOV information that we calculated during the initialisation phase.

We already know the current position of the object on the display device (we found this from our size calculation above). If we therefore keep a record of the previous screen position and a timestamp for that position, then we can calculate the velocity (pixels/s) of the object as:

$$\frac{|currOrigin - prevOrigin|}{currTime - prevTime}, \tag{4.1}$$

where $currOrigin$ and $prevOrigin$ are 2D coordinates supplied in units of pixels, and $currTime$ and $prevTime$ are given in units of seconds. This value can then be scaled into units of deg/s in the same way that the c/pixel values for spatial frequency were scaled into c/deg.

Object Eccentricity (E deg) : finally, we need a measure for the degree to which the object exists in the user's peripheral field. This is simply the angular distance between the object and the **focus point** (e.g. the point on the display which the user is gazing towards). Once again we can find this value in units of pixels and then scale this into units of degrees based upon the display FOV.

We must be careful when calculating this value: that is, we must not forget the fact that an object is not normally a point source. For example, if we measure the object's eccentricity by simply taking the distance between the focus point and the projected centre of the object, then there will be parts of the object which are closer to the focus point. As a result we run the risk of underestimating the visibility of these parts of the object.

Our measure of eccentricity must therefore be based upon the distance between the focus point and the projected point within the object which is closest to the focus point. We can approximate the latter by choosing the point on the object's smallest bounding sphere which is closest to the focus point. To improve efficiency, we could choose this value for our size calculation and hence reuse that value here (see Figure 4.4 below).

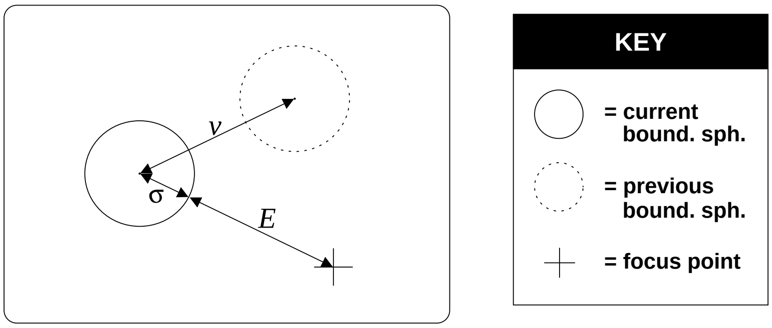


Figure 4.4: *Illustrating the pixel lengths used for the computation of object size (σ), velocity (v), and eccentricity (E). The two spheres depict the projected smallest bounding sphere of the object for the current and previous frames.*

4.4.3 Evaluating Object Frequency Content

Now that we know the size (σ), velocity (v), and eccentricity (E) of an object, we can continue with the next phase of our model. The goal of this phase is to assess the perceptual content of an object's various LOD models. This will involve using the results from the off-line stage in order to approximate the transition spatial frequencies (α_{trans}) for the object based upon its current orientation and size.

4.4.3.1 Finding Transition Spatial Frequencies for Arbitrary Orientations

We know the precise transition spatial frequencies of the object for a number of sampled viewpoints around the object (i.e. for a number of sampled orientations of the object). We wish to use this information in order to estimate the transition frequencies of the object for any arbitrary orientation.

Firstly we must resolve the orientation of the object relative to the viewpoint. This simply means finding the difference between the orientation of the 3D vector connecting the object and the viewpoint, and the orientation of the object in 3D space. This tells us the orientation of the object as perceived by the user.

We then estimate the transition spatial frequencies of the object at that orientation by interpolating between the closest sampled viewpoints around the view sphere (e.g. Renka, 1984). For example, if we used a triangular tessellation of the sphere to generate the viewpoints, then we could find the three closest sampled viewpoints and interpolate between these to find the correct value of transition spatial frequency.

Alternatively, a more efficient (but less accurate) method would be to use a point sampling technique where we find the sampled orientation which is closest to the object's current orientation and simply take this value as the result for that orientation. Of course the accuracy of this method will improve as we increase the number of viewpoints that are sampled.

4.4.3.2 Factoring Object Size into the Transition Spatial Frequencies

The values for transition spatial frequency that we have managed to compute so far are only valid for an object which is the same size as when it was sampled during the off-line stage. However, we can easily generalise our model for objects of any size by scaling our result with the relative size of the object.

Consider the example of a book sitting on a desk. If this book was then moved away until its size on your retina was exactly half the size of the original, then all of the features within that object will also be half their original size, e.g. the writing on the cover of the book. By corollary, all of the spatial frequencies within the book will have doubled (because spatial frequency is inversely proportional to size). We can therefore see that in order to incorporate object size into our model we should multiply the present transition frequency values by the fraction,

$$\frac{\sigma_0}{\sigma}, \tag{4.2}$$

where σ is the current size of the object on the display device (pixels) and σ_0 is the size of the object when it was sampled during the off-line stage (pixels).

We now know the instantaneous values for transition spatial frequency (α_{trans}) within an object. These values tell us the threshold frequencies at which we can select the different LOD models for the current view of the object. All we require now is the ability to ascertain the degree of spatial frequency which the user can perceive in the object under its current viewing conditions.

4.4.4 Evaluating Object Visibility

The goal of this phase is to estimate the **highest visible spatial frequency** (α_{vis}) that we expect the user to be able to perceive in an object. This is done by applying our model of contrast sensitivity using the values of object eccentricity (E) and velocity (v) that we have already calculated.

4.4.4.1 Applying our Visual Acuity Model

Earlier, in Section 3.2.3 (Page 93), we described visual acuity in terms of our contrast sensitivity model using the relation:

$$H(\alpha, v, E) - 1 = 0, \tag{4.3}$$

where the function $H(\alpha, v, E)$ is defined in Equation 3.21. We wish to use this relation by supplying values for v and E and then subsequently solving the equation for α, thus giving the highest spatial frequency which is visible under those spatiotemporal conditions.

Unfortunately we have already noted that an analytical approach to solving this relation for α would be computationally involved, and that an iterative method would be more elegant (see Section 3.2.3, page 93). We therefore advocate an interval halving algorithm here. This proceeds by instantiating the v and E parameters with the computed values for object velocity and eccentricity, respectively. Then an initial value for α is taken and this estimate repeatedly refined until the resulting value of $H(\alpha, v, E) - 1$ is within a desired tolerance of zero. A sufficiently large initial value for α should be chosen so that the highest root of $H(\alpha, v, E) - 1 = 0$ is always found, e.g. the highest spatial frequency that our visual system can resolve, α = 60 c/deg (Campbell and Gubisch, 1966).

Naturally such an iterative calculation should be avoided during a real-time simulation for performance reasons. So we propose that values from this process should be precomputed when the simulation is first started and the results stored in a look-up table (LUT) for optimum performance.

4.4.4.2 Defining the Highest Displayable Spatial Frequency

The above model would be sufficient for evaluating the visibility of objects in the real world; however we must take into consideration the fact that the user is viewing a computer display, and so the angular resolution of that display will limit the size of detail which they can be exposed to. We can incorporate this factor into our model by introducing the notion of a **Highest Displayable Spatial Frequency**, ξ. This characterises the highest frequency which can be displayed by an output device, and will of course depend upon the field of view and pixel resolution of the device.

Referring to Section 3.1.5.1 (Page 76), we recall that a single pixel can be considered to be half a contrast cycle. Therefore the number of contrast cycles that can be displayed on a device is equal to half its pixel resolution. This can be transformed into a value of spatial frequency once we know the FOV of the display using the following simple equation:

$$\xi = pixelRes/(2 \times angularRes), \tag{4.4}$$

where $pixelRes$ represents the number of pixels in the display and $angularRes$ represents the angular field of view of the display (in degrees).

As part of the rationalisation of our model, we resolved to ignore the orientation of spatial frequencies in our implementation. We can therefore describe our final definition of highest displayable spatial frequency as the largest value between the horizontal and vertical resolutions of the display device, i.e.

$$\xi = \max\left(\frac{horizPixels}{2 \times horizFOV}, \frac{vertPixels}{2 \times vertFOV}\right), \tag{4.5}$$

where $horizPixels$ and $vertPixels$ are the horizontal and vertical pixel resolutions of the display, and $horizFOV$ and $vertFOV$ are the horizontal and vertical angular resolutions of the display. (Refer to Equations 3.2 and 3.3, Page 77, on how to calculate values for $horizFOV$ and $vertFOV$ for a desktop display.)

4.4.4.3 Completing our Visual Acuity Model

Now that we can define the highest displayable spatial frequency for a display device, we can incorporate this factor into our model. We do this by simply thresholding our present value for highest visible frequency (calculated from our contrast sensitivity model) with the appropriate value of ξ (calculated from the display resolution), i.e. we take the smallest of these two values (see Figure 4.5). This means that our model will now tell us the highest visible spatial frequency that can be perceived at a particular velocity and eccentricity, taking into consideration the viewing conditions and display device limitations. We shall refer to this value as α_{vis}.

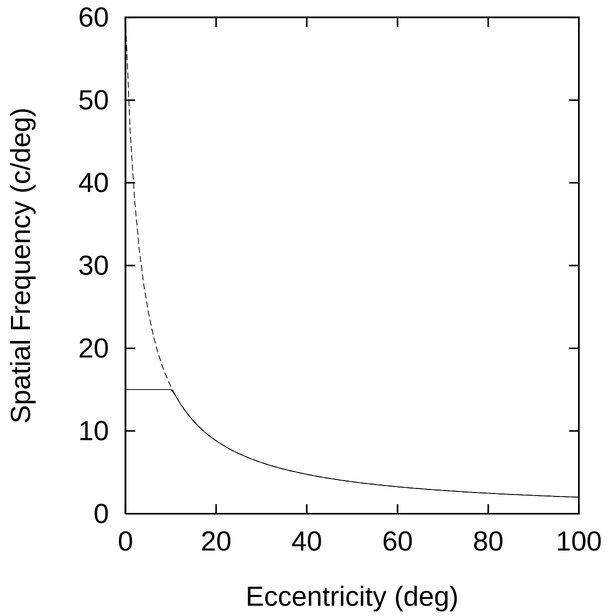


Figure 4.5: *The effect of incorporating maximum displayable spatial frequency into our model. The curves show the decline of spatial frequency sensitivity with eccentricity only (using Equation 3.20 with M_0 = 60 c/deg). The broken line illustrates the original model, whilst the overlayed solid line indicates the effect of thresholding the model with ξ = 15 c/deg.*

It may be instructive to note that our term, highest displayable spatial frequency, is merely a simplification of the display device's **modulation transfer function** (MTF). The MTF is a measure of a display's ability to maintain the contrast of a signal as a function of its spatial frequency (Evans, 1990), i.e. in effect it is the electronic equivalent of our biological system's contrast sensitivity function (CSF). The general operation would therefore be to threshold the CSF using the MTF. However, because we are only dealing with visual acuity here—i.e. the uppermost point of the CSF—we only need use the equivalent uppermost point of the MTF. It is this point on the MTF which we have called highest displayable spatial frequency (ξ).

4.4.5 Selecting the Optimal LOD

As a result of the preceding calculations, we now know the highest spatial frequency that the user should be able to perceive in an object (α_{vis}) and the effective transition frequency (α_{trans}) between each level of detail of that object. It is now a simple matter to find the best LOD to use for that object, e.g. the lowest LOD such that the α_{trans} value for the immediately higher LOD is greater than the limit of vision for the object, α_{vis}. That is, the least complex model such that the smallest visual change that it will entail will be below the user's threshold of vision.

Here we make the assertion that the value of α_{trans} associated with model l is for the transition between models $l-1$ and l, where model $l-1$ is defined to be less complex than l. Of course, if all models' α_{trans} value are below α_{vis}, then the highest LOD should be selected (because this means that selecting any model other than the highest LOD will produce a visible switch).

The following section of C++ source code illustrates the implementation of the LOD selection routine, given the values for α_{trans} and α_{vis}. These two values are referred to in the code segment as the array `object.SFtrans[]` and the scalar `object.SFvis` respectively.

```
int LOD = object.highestLOD;
while ( LOD > object.lowestLOD && object.SFtrans[LOD] > object.SFvis )
   LOD--;
object.setLOD( LOD );
```

It is worth noting one significant optimisation which can be made here, now that we have covered all the appropriate material. This follows from the observation that we may not need to calculate the transition spatial frequency between every LOD. For example, if we eventually select the highest LOD then we only need to know the transition frequency associated with that model. In the above code segment we can therefore replace the array look up on the previously computed α_{trans} value (`object.SFtrans[LOD]`) with a class method that actually performs the transition frequency calculation *in situ* and returns the resulting α_{trans} value. This way, if the algorithm selects the highest LOD and so exits the while loop on the first iteration, then we only compute one transition frequency.

4.4.6 Focussing the On-line Stage

Figure 4.6 illustrates how the various phases of the on-line scheduler can be integrated into the standard **Sense–Process–Display** loop of a typical VR system. The position of the scheduler within the Process stage is important in order to gain the optimal performance. It must follow any processing which updates the location and velocity of each object (including the viewpoint); but it should precede any subsequent processing (e.g. collision detection, dynamics equations, etc.) so that this will be applied to the actual model about to be displayed.

To collate all the material that we have just discussed with regards to the on-line stage, we will now take all of the phases indicated by the right flow diagram in Figure 4.6 and outline the major operations involved in each of these.

(a) **Find Object's Size, Velocity, and Eccentricity** : the first action we must perform is to calculate the size of an object in screen space (pixels), its angular velocity based upon its previous position (deg/s), and the eccentricity of the point on the object's smallest bounding sphere which is closest to the focus point (deg). In the case of the latter two attributes, we first calculate these in terms of pixels and then transform these into values in terms of degrees based upon the display FOV and resolution.

(b) **Evaluate Transition Frequencies** : this phase attempts to encapsulate the spatial frequencies within each of the object's levels of detail. We have reduced this representation to a single viewpoint-dependent and

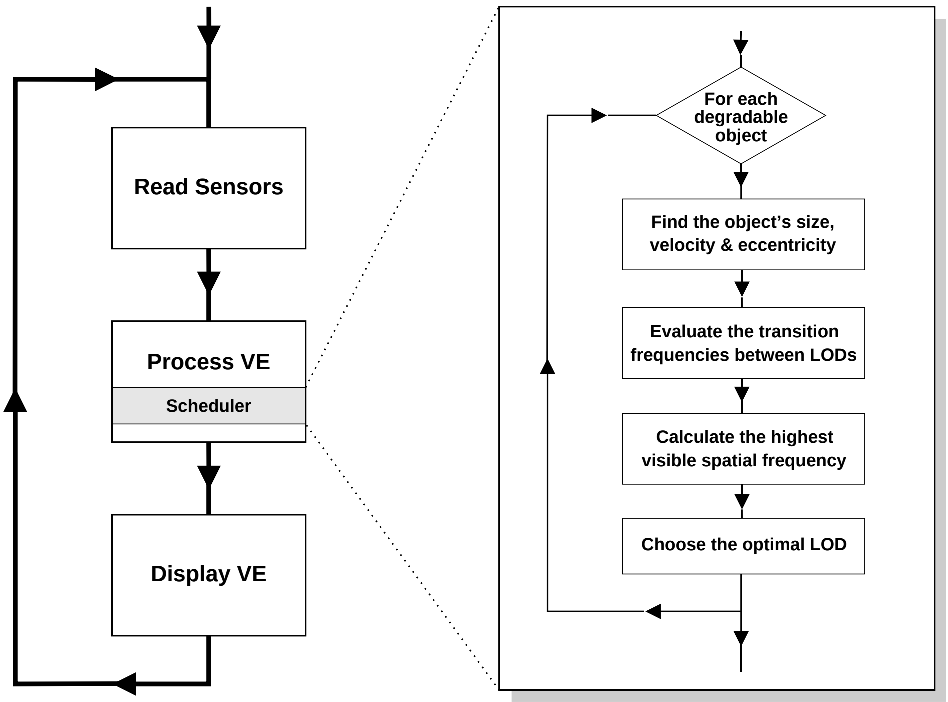


Figure 4.6: *Illustrating how our LOD scheduler (right-hand flow diagram) fits into the main loop of a typical VR system (left-hand flow diagram).*

size-dependent value (α_{trans}) between each LOD. This value defines the threshold spatial frequency that describes the size of the smallest feature which differs between successive LODs. We calculate this value for the current view of the object by using the sampled values from the off-line stage to estimate each LOD's α_{trans} for the perceived orientation, and then scale these results based upon the relative size of the object on the display device.

(c) **Calculate Highest Visible Frequency** : this process attempts to estimate the degree of spatial detail which the user can perceive in the object, based upon its current velocity and eccentricity. That is, the highest spatial frequency (α_{vis}) that we would expect them to be able to resolve in the object. This value is calculated from our model of visual acuity (which in turn is derived from our contrast sensitivity model), and the spatial frequency limit of the display device.

(d) **Choose Optimal LOD** : finally, we can use the value of α_{vis} and the values for α_{trans} between each LOD to select the optimal LOD for that object such that the user should not be able to perceive any visual change. This is the lowest LOD where all visual changes between this model and the highest LOD are below the user's threshold of vision.

It should be clear how our model manages the optimal LOD of an object based upon its velocity and eccentricity, however it may be useful to clarify why our model also finds the optimal LOD to display based upon the object's size on the display device. To do this, we will take the example of a large blackboard with a piece of chalk resting on its ledge. Let us say that the chalk is displayed such that it is 10 c/deg long and that the highest displayable spatial frequency of the output device is 15 c/deg. If the blackboard is moved away so that it becomes half the original size, then the chalk will be have twice the spatial frequency, i.e. 20 c/deg. Ordinarily this may still be visible to the human eye, but because the display's maximum limit is 15 c/deg, we know that the chalk is too small to be displayed. Effectively, by incorporating the notion of a highest displayable spatial frequency (ξ) into our contrast sensitivity model we gain the ability to calculate when a feature projects to less than a single pixel.

4.5 Summary

This chapter has introduced a general model for perceptual LOD, and subsequently described its efficient implementation. We have seen that our model must be split into two principal stages: an off-line and an on-line stage. The off-line stage is where the various levels of detail for an object are generated, and where these are analysed to assess their spatial frequency content. The on-line stage is composed of a scheduler that selects the optimal LOD to use at each update of the simulation. This is done through a process of estimating the actual spatial frequency of the object from the off-line data, and predicting the visual acuity of the user with regard to the object.

We proposed several simplifications to our model in order to produce a feasible real-time implementation. However, none of these simplifications should be taken as inexorable constraints. It would be possible to increase the complexity of the implementation (e.g. incorporate and act upon orientation information) if more compute time were available, or if the real-time constraint were lifted (e.g. for a passive animation sequence).

We will defer discussion of the general issues surrounding perceptual LOD until a later chapter. However, the particular approach that we have adopted raises a number of issues which are worth identifying here. The following points therefore provide a discussion for the most prominent of these issues.

- **Orthogonality** : each of the three components of our model can be used individually or in any desired combination. For example, we can disable the modulation of detail with respect to eccentricity by simply instantiating all eccentricity values to 0 deg. Our model will then still function correctly, but will only modulate detail based upon the size and velocity of an object. Similarly we can disable velocity optimisations by instantiating all velocity values to 0 deg/s, and disable size optimisations by defining the relative size of an object (Equation 4.2) as 1.0.

- **Eye Tracker Support** : we have already commented that our visual perception is based upon the attributes of stimuli projected onto our retinae, and that to be completely accurate we should therefore incorporate an appropriate eye tracking technology into our system.

  Our model is sufficiently general to encompass this facility, but not require it. That is, the notion of a focus point can be used to represent the position of a user's gaze on screen, or it can simply be defined as a fixed location if eye tracking is not available (e.g. the centre of the screen).

- **Resolution Adaptability** : we sample each object at a specific resolution to find its spatial frequency content. This therefore raises the question of what happens if the simulation is run at a different resolution to that of the off-line sampling stage?

  In the first case, if different resolutions are chosen for the on-line and off-line stages, but these have the same aspect ratio, then this will cause no problems. For example, we could sample objects at a resolution of 640×480 pixels and then use a simulation resolution of 1280×960 or 320×240. This is possible because simply changing the resolution of the display does not alter the degree of detail which is displayed within an object at a certain size: it simply means that we can display objects at a larger (or smaller) screen size.

  If the aspect ratio between the on-line and off-line resolutions differ then we could simply scale all transition spatial frequencies appropriately or, more accurately, we could repeat the spatial frequency sampling process for that on-line resolution.

On a related issue, if we change the FOV of the display but retain the same pixel resolution then this will also cause no problems. This is because our model automatically compensates for any FOV by delaying the conversion from c/pixel values into c/deg until the simulation is actually begun.

- **Limitations of the Model** : flexibility and accuracy have been prime objectives during the development of our model. However, there comes a point when these two factors must be traded off against each other. The biggest limitation which results from the particular approach that we have adopted is that we cannot compensate for any environmental changes during a simulation (e.g. changes in lighting, fog, time-of-day, etc.). This is because our assessment of an object's perceptual content is based upon data which are collected off-line.

  As we have already noted, this is not a severe limitation because we can simply ensure that we collect an object's spatial frequency profile under ideal conditions (e.g. high illumination, no fog, etc.). This ensures that our model will function accurately under all environmental conditions, although it may not function as optimally under substantially degraded conditions.

Up to this point our discussion has been mostly conceptual in content. As a result we are left pondering a number of questions regarding the implementation of our model. For example, to what extent can our model reduce lag in a VR system? How accurate is our visual acuity model? Does our model succeed in its goal of modulating LOD with no perceptual artifacts? These and other issues will be addressed in the next chapter which presents an evaluation of our implementation for perceptually modulated level of detail.

Chapter 5

Results

'To tell us that every species of things is endowed with an occult specific quality by which it acts and produces manifest effects, is to tell us nothing'

(Sir Isaac Newton, *Opticks*)

A prototype implementation was developed for the perceptually-based LOD model that we have just described. This chapter will consequently present an empirical evaluation of the prototype system. To fulfil this objective, a test suite of experiments was devised to evaluate:

1. the accuracy of the model,
2. the effects of the model on the user, and
3. the effects of the model on system performance.

These three goals were achieved respectively via a number of psychophysical studies, user studies, and computational studies. This chapter is therefore structured into three sections, covering each of these topics in turn.

5.1 Psychophysical Studies

This section describes a number of psychophysical experiments which were performed as part of the evaluation suite. Psychophysics is the branch of psychology which is concerned with establishing quantitative relations between physical stimulation and perceptual events. Therefore, the following experiments attempt to analyse the effect of our LOD modulation on the user's perception of the scene. That is, to assess the accuracy of our model in its goal of modulating detail without the user perceiving any visual change.

As with most psychophysical experiments, the stimuli used must be very simple. This is necessary in order to isolate the experiment to the particular condition under analysis, and to reduce the influence of any extraneous or obfuscatory variables.

We will evaluate the three facets of our perceptual model independently so that we may gain some insight into the efficiency of each individual component. We will therefore look at how well our model predicts the visibility of stimuli based upon their size, their eccentricity, and their velocity.

5.1.1 Object Size

5.1.1.1 Objective

This first experiment aims to establish how well our perceptual model can predict the optimal LOD switching thresholds based upon the size of an object in screen coordinates. That is, to efficiently circumvent the popping effect which is often associated with traditional size or distance LOD techniques.

5.1.1.2 Method

For each object's LOD, the optimal switching threshold was determined in terms of object size. This was then compared with the estimated results from our perceptual model. The following definitions were made:

- The size of an object was defined as the length of the radius of the object's smallest bounding sphere once it had been projected into screen coordinates.
- The 'optimal' switching threshold was evaluated using an automated brute force method as follows: the LOD model being tested and the next highest

LOD were displayed under identical conditions. An interval halving algorithm was used to converge on the shortest distance at which there was no discernible difference between the display of the two LOD models, i.e. if the two images were perceptually dissimilar, then the object was moved further away (object becomes smaller). If the two images were perceptually identical, then the object was moved closer (object becomes larger). The distance translated was halved at each iteration and the process terminated as this step size tended towards zero.

- The question of how to decide whether two images are 'perceptually identical' is an interesting one. As Rushmeier *et al*. (1995) note, the simplest procedure would be to conclude that two images are identical if and only if every corresponding pair of pixels in each image is identical. This will of course provide a somewhat over cautious measure because very small differences between a pixel's colour may not be discernible.

  The metric which was adopted here was therefore to define two images as being perceptually identical if and only if every corresponding pair of pixels were perceptually identical. We used the CIELUV colour difference equation as a measure of whether two pixels were perceptually identical, i.e. images $I_1(x, y)$ and $I_2(x, y)$, both of dimensions $M \times N$, were deemed perceptually identical if:

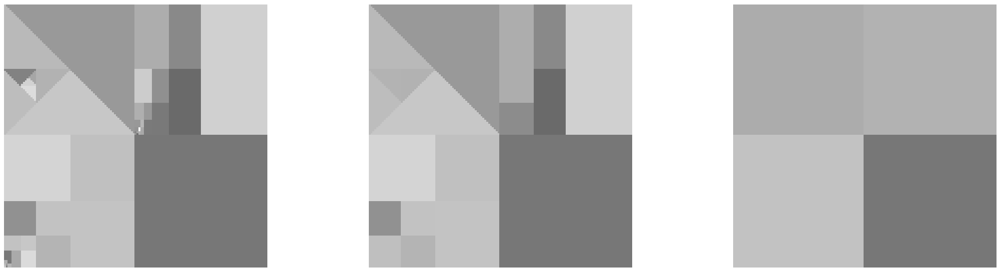
$$\Delta E^*_{uv}(x, y) < \epsilon, \quad \text{where} \quad 0 \leq x < M, 0 \leq y < N, \tag{5.1}$$

  where $\Delta E^*_{uv}(i, j)$ represents the CIELUV colour difference between pixels $I_1(i, j)$ and $I_2(i, j)$. We chose a threshold value of $\epsilon = 5$ CIELUV units to provide a more stringent measure of colour difference than we have used previously. The value of $\Delta E^*_{uv}(x, y)$ was size corrected for a stimulus size of 0.1 degrees using Silverstein and Merrifield's (1985) formulae (this represented a display region of $\sim$2 pixels). Finally, the computer predicted thresholds were confirmed by visual inspection.

- The estimated results from our perceptual model were calculated in the manner described in the previous chapter. Each LOD was displayed so that it occupied the entire display (a display resolution of 700 $\times$ 700 pixels was used). The size of the object and all of its spatial frequencies were recorded in each instance, and then transition spatial frequencies were calculated to characterise switching conditions for each LOD.

5.1.1.3 Results

Two objects were used for this experiment. The first object was a very simple 2D test card which contained a number of randomly coloured triangles, squares, and rectangles at various scales. This object was chosen to investigate our model's basic ability to extract perceptual features and to subsequently predict their visibility. Figure 5.1 illustrates three different levels of detail for the test card object.



(a) (b) (c)

Figure 5.1: *The test card pattern at three different levels of detail. Eight levels of detail were generated for this model. The figure shows models 8, 4, and 2, respectively.*

In total, eight levels of detail were produced for the test card object. All of the lower levels of detail were generated automatically from the highest LOD using the perceptually-driven LOD generation paradigm advocated in Section 3.3. The results from the experiment are presented in Figure 5.2.

The second object was a simple 3D representation of a die. This was used in order to test our model with a more realistic, three-dimensional stimulus; and also to show the model performing on different viewpoints of the same stimulus. The die object consisted of three levels of detail. Sample results are provided in Figure 5.3.

5.1.1.4 Discussion

From the results of this experiment we can observe that the predicted switching thresholds compare very favourably with the optimal switching thresholds.

We note that there is an occasional discrepancy between the predicted and optimal thresholds. This could be due to small inaccuracies in the brute force

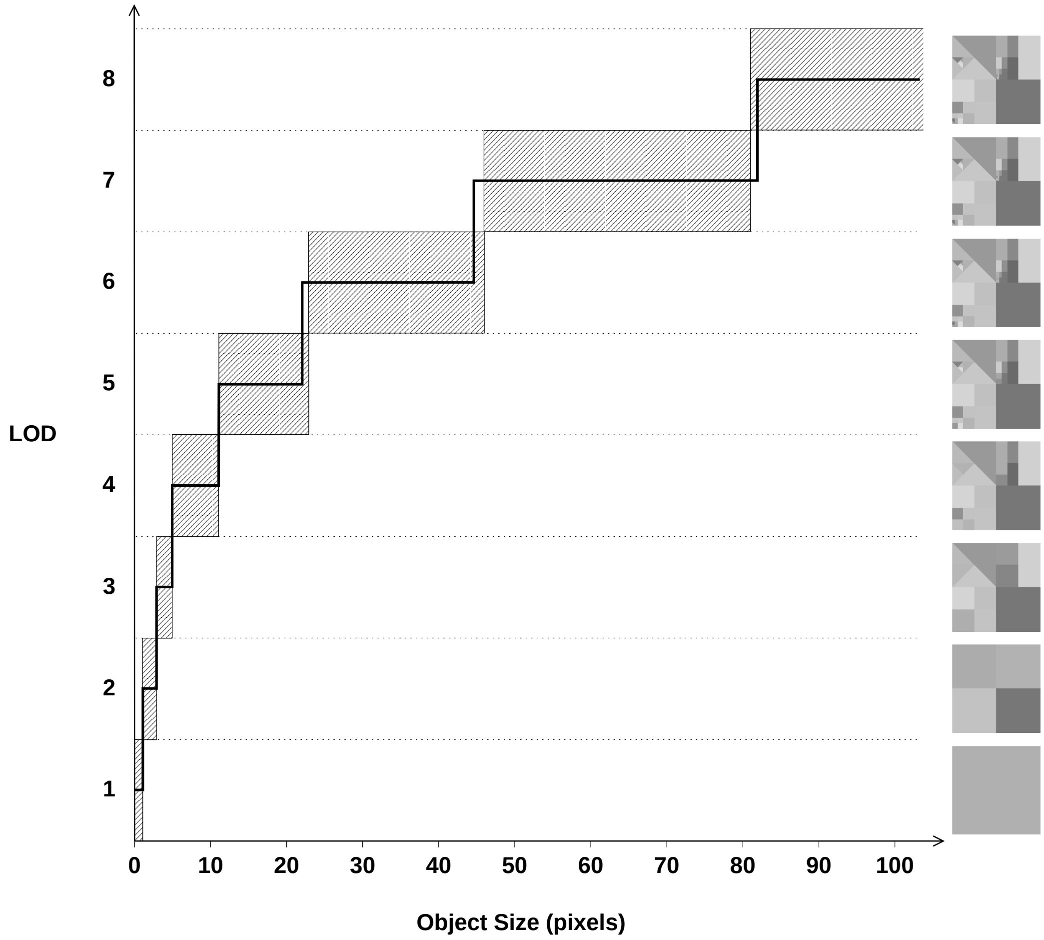
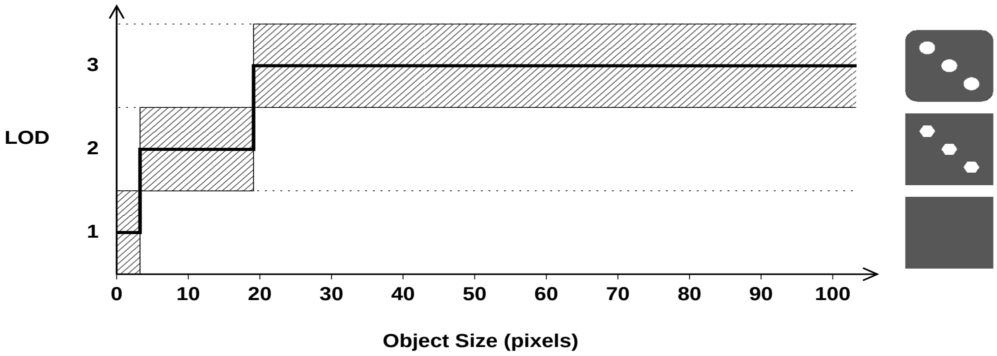


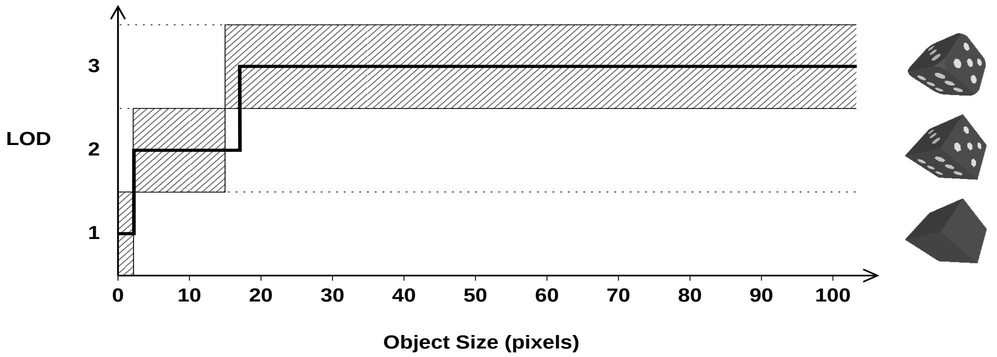
Figure 5.2: *Actual and projected switching thresholds (based upon object size) for a test card pattern. The shaded regions represent optimal switching ranges as found by the brute force algorithm. The solid line represents the projected switching thresholds estimated from the spatial frequency profile (object analysed at size = 493 pixels).*

algorithm and/or the spatial frequency extraction process. However, having noted this, the discrepancies are only in the order of 1 to 2 pixels: around 0.2% of the display resolution. This is certainly an acceptable error margin and the author can attest that no significant or distracting popping effects were experienced.

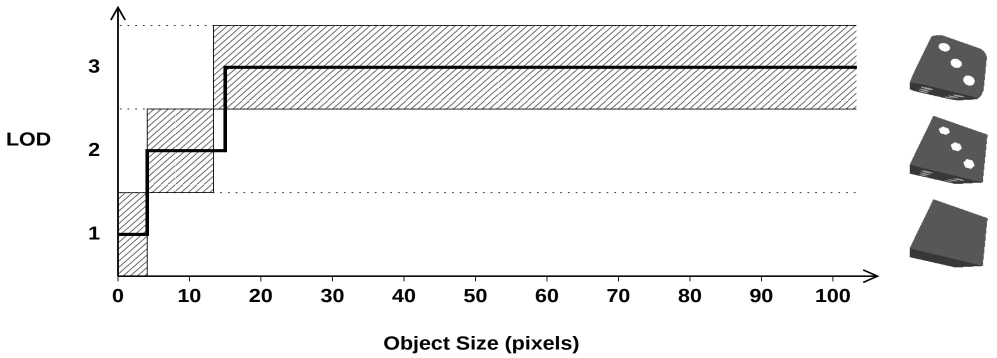
We can therefore conclude that our model is able to efficiently and accurately locate optimal switching sizes for different LODs of an object with minimal visual artifacts.



(a)



(b)



(c)

Figure 5.3: *Actual and projected switching thresholds (based upon object size) for a model of a die at three orientations. The shaded regions represent optimal switching ranges as found by the brute force algorithm. The solid line represents the projected switching thresholds estimated from the spatial frequency profile (object analysed at size = 360 pixels).*

5.1.2 Object Eccentricity

5.1.2.1 Objective

This next experiment was devised to assess the effective threshold eccentricity for a number of 2D, aperiodic stimuli at a fixed contrast and spatial frequency. This was performed in order to establish how well our perceptual model can predict the user's ability to resolve detail in their peripheral field.

5.1.2.2 Overview

The user was presented with a number of simple stimuli at various points in their peripheral field. The experiment was devised to locate the user's threshold eccentricity for a stimulus of fixed size and contrast. This was then compared with the predicted threshold from our perceptual model.

The 2AFC Method

The design of this type of experiment is crucial. It has been shown that presenting a single stimulus to a subject and asking whether they can see it—replying 'yes' or 'no'—is a flawed approach. This is because most subjects appear to have a predisposed readiness to report positive responses (Lamming, 1991a). Therefore, the generally recommended method for determining signal detection thresholds is the **2 Alternative Forced Choice** (2AFC) method. This involves presenting the subject with the test stimulus in one of two successive observation intervals and asking them to decide which interval it occurred in, guessing if necessary (Lamming, 1991b). For example, presenting the stimulus to the left or to the right of the user's gaze.

When the stimulus is above threshold then the user will always identify the stimulus interval correctly. However, when the stimulus is below threshold the user will have to guess and their performance will drop to chance level ($\sim$50%). Again, when guessing, the observer may exhibit a predisposed tendency to choose the first (i.e. left) interval. It has been shown that this can be countered by randomly varying the stimulus interval (Simpson, 1989).

The Staircase Method

There are two principal procedures for executing a 2AFC experiment. These are the **constant stimuli** method and the **staircase** method. In our present context, the former would mean selecting a number of eccentricities (normally five) and only testing the subject with these values. Alternatively, the staircase method adaptively selects new eccentricities to analyse based upon the subject's previous responses; eventually converging around the threshold value. The staircase methodology is often considered the more efficient and robust approach (Levitt, 1971), and so we will adopt this technique.

The operation of the staircase method (also known as the **Transformed Up–Down** method) can be described as follows (Wetherill and Levitt, 1965):

1. Choose an initial eccentricity (our best *a priori* estimate, e.g. the predicted result from our model).
2. If the subject gets two observations correct, then make the next condition more difficult (i.e. larger eccentricity).
3. If the subject gets one observation wrong, then make the next condition more easy (i.e. smaller eccentricity).
4. The experiment terminates after 6–8 reversals, where a reversal is defined as the turning point after a sequence of stimulus changes in the same direction.

The final result is then found by averaging the values at each of the reversals. The asymmetric nature of the algorithm (i.e. two correct responses induce an increase in difficulty, but only one incorrect response is required to decrease the difficulty) is designed to locate the eccentricity at which, statistically, 70.7% positive responses are obtained. This is normally taken as a measure of detection threshold.

5.1.2.3 Method

Stimuli. The stimuli were displayed at full contrast (a white stimulus on a 2 deg black patch) on a mid-grey background. The display consisted of two black patches, one to the left and one to the right of the crosshair (both patches were equidistant from the crosshair). The stimulus was displayed randomly in either the left or the right interval and retained the same contrast and spatial frequency throughout the experiment; only varying in the eccentricity at which it was displayed to the user. Figure 5.4 illustrates the stimuli used for this experiment.

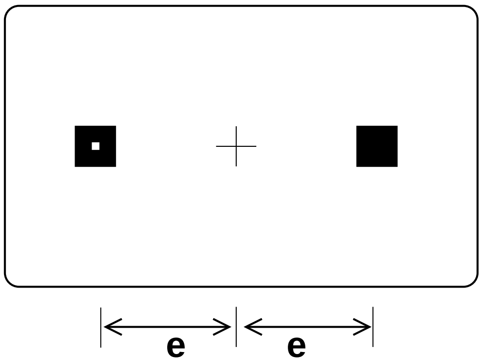


Figure 5.4: *The display used for the eccentricity 2AFC test. The stimulus was presented at eccentricity* e*, in either the left or the right patch (shown here on the left).*

The experiment was run for a number of different sized stimuli (1 and 2 pixels square) at a number of different viewing distances (10, 20, 30, 40, 50, and 60 cm). Overall, seven different spatial frequencies were analysed: 2.52, 5.04, 7.11, 9.50, 12.05, 14.68, and 17.36 c/deg.

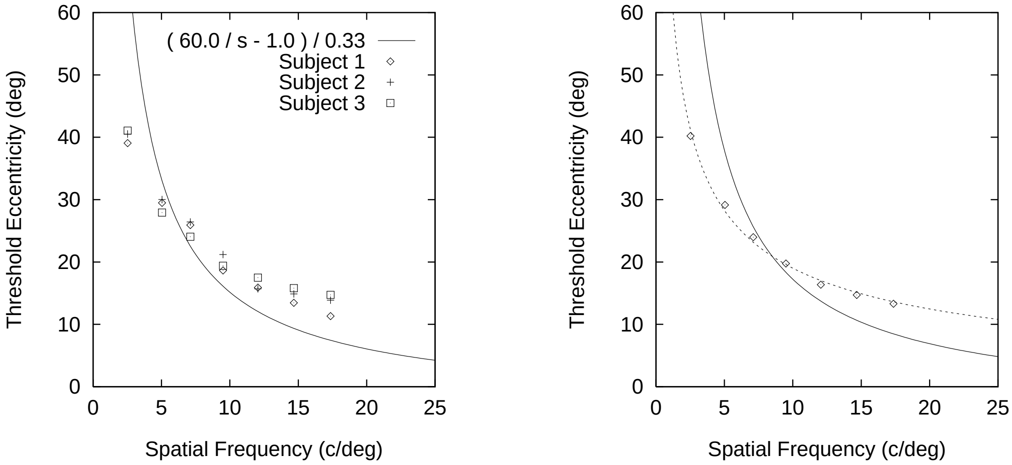
A staircase algorithm was employed to converge upon the user's threshold eccentricity for each stimulus. The initial step size was 1 deg. This was reduced to 0.5 deg after the first reversal, and further reduced to 0.25 deg after 5 reversals. The experiment terminated after 10 reversals.

Procedure. Three subjects, with normal or corrected-to-normal vision, performed each experiment twice. All subjects were unpaid male postgraduate students in the Department of Computer Science at the University of Edinburgh.

Each subject fixated on a crosshair displayed at eye level on the computer monitor. Eye movements were not monitored, but a chin rest was used in order to restrict head movement and to preserve the viewing distance. The subject was asked to chose the correct interval by pressing one of two buttons on the keyboard. They could abort any observation with another button if desired and this data would not be included in the final result.

Subjects were allowed a number of test runs beforehand in order to acquaint themselves with the experimental technique. The final threshold figures were found by averaging each pair of results from all subjects.

5.1.2.4 Results



(a) **(b)**

Figure 5.5: *Results of the eccentricity experiment. (a) presents the individual responses of each of the three subjects, while (b) presents the averaged response for all subjects (the data points) and the best fit curve to this average response (the dotted line: Equation 5.3). The continuous line in (a) and (b) represents the predicted response from our model, given by Equation 5.2.*

The results from this experiment are presented in Figure 5.5. In order to calculate the predicted threshold response, we used our cortical magnification factor, defined by Equation 3.20. This can be written in terms of spatial frequency (α) as:

$$E = (M_0/\alpha - 1)/0.29, \tag{5.2}$$

where $M_0 = 60$ c/deg: the highest spatial frequency which can be resolved at the fovea (Campbell and Gubisch, 1966).

The experiment produced results which were in the correct order of magnitude, but somewhat displaced from our predicted curve. A best fit curve was therefore calculated which represented the data more closely. This curve is defined in Equation 5.3 below, and plotted in Figure 5.5(b).

$$E = (21.2/\alpha^{0.5} - 1)/0.3. \tag{5.3}$$

We can use this curve to formulate a new model for our cortical magnification factor by first rewriting Equation 5.3 in terms of spatial frequency, e.g.

$$\begin{aligned} \alpha &= (21.2/(0.3E + 1))^2 \\ &= 449.44/(0.3E + 1)^2, \end{aligned} \tag{5.4}$$

and then normalising this so that a value of 1.0 is returned at the fovea and values tending towards 0.0 are returned for highly eccentric locations. We have already reported that the highest spatial frequency visible at the fovea is 60 c/deg. Using Equation 5.3 we find that our best fit curve does not go below this spatial frequency until an eccentricity of 5.79 deg is exceeded. We can therefore present our new equation for cortical magnification as follows. This equation is plotted in Figure 5.6.

$$M = \begin{cases} 1.0, & \text{when } E \leq 5.79 \\ 449.44/((0.3E + 1)^2 \times 60) & \\ = 7.49/(0.3E + 1)^2, & \text{when } E > 5.79. \end{cases} \tag{5.5}$$

5.1.2.5 Discussion

The results from this experiment are very encouraging. They display a clear and smooth decline in spatial frequency sensitivity with increasing eccentricity. The results for all subjects lie consistently within 2 deg of the average result, and mostly within 1 deg. Given the inherent variability of individuals' vision systems, and taking into consideration any experimental error, this is a very good result. However, it is apparent that the empirical data, although in the correct order of magnitude, does not exactly match the theoretical threshold. This could be due to inaccuracies in our theoretical model (Equation 3.20), however, the most likely reason for the discrepancy is that, being based solely on retinal ganglion cell distributions, Rovamo and Virsu's (1979) model may not encapsulate the total processing performed by the entire visual system.

We therefore produced a best fit curve (Equation 5.3) which better models the observers' decline of peripheral sensitivity, and subsequently formulated a new model of cortical magnification for our application (Equation 5.5).

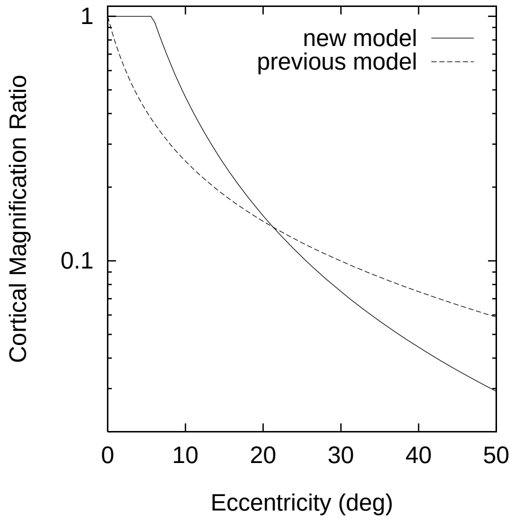


Figure 5.6: *Comparison between our new and previous definitions of the cortical magnification factor, M. The solid curve represents our new empirically derived model (Equation 5.5), whilst the broken curve represents our previous definition (Equation 3.20).*

5.1.3 Object Velocity

5.1.3.1 Objective

This final psychophysical experiment was devised to assess various subjects' threshold velocity for a number of 2D, aperiodic stimuli at a fixed contrast and spatial frequency. This was performed in order to establish how well our perceptual model can predict the user's ability to resolve detail based upon the velocity of a stimulus.

5.1.3.2 Method

Stimuli. Two black 2 deg patches were animated vertically past the observer's viewpoint (at 72 Hz) with a constant angular velocity; one to the left of the crosshair and one to the right. A white (full contrast) stimulus was randomly displayed in either the left or right interval during each trial and the observer had to chose which interval they thought the stimulus had appeared in. The stimuli were always presented at a horizontal angular distance of 2.5 deg from the crosshair in order to minimise the effect of eccentricity on detection (recall from Section 1.4.4.2 that our vision is maximally sensitive within the central $\sim$5 deg region. Also, from our results in Figure 5.6 we can observe that within 0 to

5.79 deg all potential stimuli would be visible under static conditions). Figure 5.7 illustrates the display used for the experiment.

A 2AFC staircase method was adopted as before. The initial step size for the staircase algorithm was 5 deg/s. This was reduced to 2.5 deg/s after the first reversal, 1 deg/s after 3 reversals, and 0.5 deg/s after 6 reversals. The experiment terminated after 10 reversals. Six different spatial frequencies were analysed: 5.04, 7.11, 9.50, 12.05, 14.68, and 17.36 c/deg.

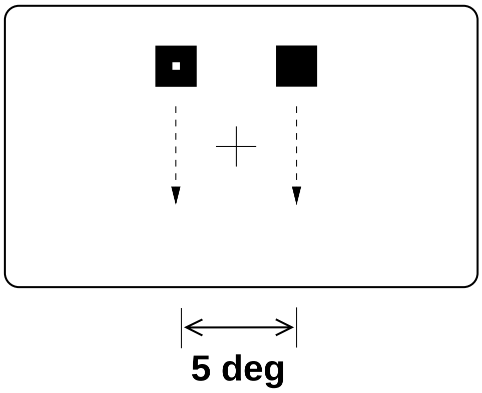


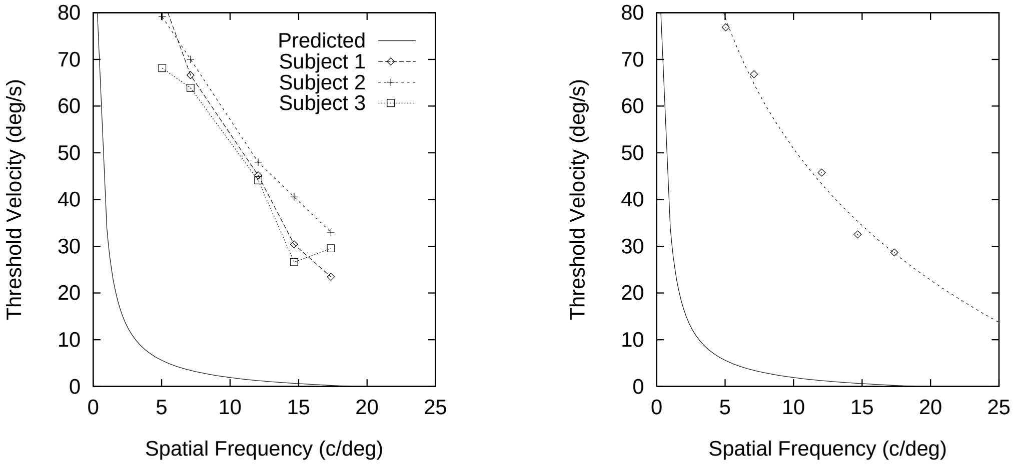
Figure 5.7: *The display used for the velocity 2AFC test. Two patches moved past the observer's fixation point at a constant angular velocity, with the stimulus being present in either the left or the right patch (shown here on the left).*

Procedure. The same three subjects who performed the previous experiment were solicited for this experiment. Each subject fixated upon a crosshair, positioned at eye level, with their head movement constrained by a chin rest. Once again, subjects were allowed a number of test runs beforehand in order to acquaint themselves with the experimental technique.

The experiment was performed twice for each spatial frequency. The final threshold figures were found by averaging each pair of results from all subjects.

5.1.3.3 Results

The results of the velocity experiment are presented in Figure 5.8. In order to calculate the predicted threshold response, we used our spatiotemporal contrast sensitivity model (Equation 3.15) and found the upper root of the relation, $G(\alpha, v) - 1 = 0$, using an interval halving algorithm.



(a) **(b)**

Figure 5.8: *Results of the velocity experiment. (a) presents the individual responses of each of the three subjects, while (b) presents the averaged response for all subjects (the data points) and the best fit curve to this average response (the dotted line: Equation 5.6). The continuous line in (a) and (b) represents the predicted response from our model.*

As can be observed from the data in Figure 5.8, the results which were obtained are substantially deviant from our theoretical model. A best fit curve was therefore calculated for the experimental data (using a computer graphing package). An exponential curve fit was used in order to retain as much as possible the general characteristics of Kelly's (1975) abstract model (see Section 3.2). The resulting curve is presented in Equation 5.6 below, and plotted in Figure 5.8(b).

$$v = 119.353 \times 10^{-0.036\alpha}. \tag{5.6}$$

We can use this curve to help us formulate a new spatiotemporal threshold model. This requires that we find the corresponding form of Equation 5.6 in terms of spatial frequency, α. That is,

$$\begin{aligned} \alpha &= \log_{10}(v/119.353)/-0.036 \\ &= (\log_{10}(v) - \log_{10}(119.353))/-0.036 \\ &= -27.78\log_{10}(v) + 57.69. \end{aligned} \tag{5.7}$$

Once again we can note that this equation only goes below 60 c/deg when $v > 0.825$ deg/s. We can therefore present the final definition for our spatiotemporal threshold model (for $v \geq 0$ deg/s) as follows. This equation is plotted in Figure 5.9 along with the previous model for comparison.

$$\alpha = \begin{cases} 60.0, & \text{when } v \leq 0.825 \\ -27.78 \log_{10}(v) + 57.69, & \text{when } v > 0.825. \end{cases} \tag{5.8}$$

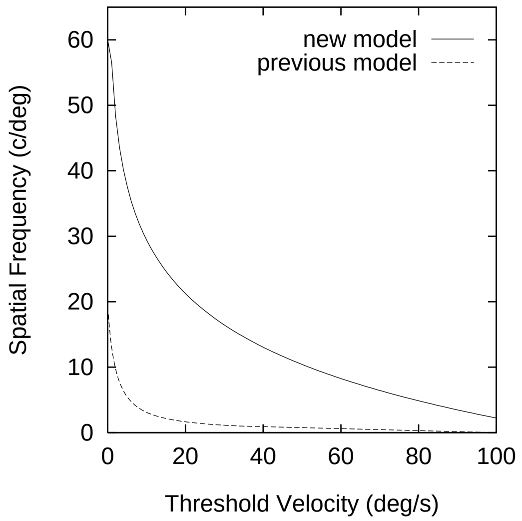


Figure 5.9: *Comparison between our new and previous definitions for the spatiotemporal threshold surface. The solid curve represents our new empirically derived model (Equation 5.8), whilst the broken curve represents our previous definition (using Equation 3.15).*

5.1.3.4 Discussion

The results from this experiment are not as smooth as those obtained from the eccentricity experiment, or as consistent with the theoretical response. However, this was anticipated given the discussion in Section 3.2.4 on the problems of defining a standard spatiotemporal threshold surface. The most probable cause for the observed discrepancy is due to the fact that we are using very localised stimuli (in terms of the field of view occupied), whereas vision scientists normally deal with extended stimuli which fill a large proportion of the FOV. Also, the experiment was more complicated, both for the author to devise, and for the subjects to perform. In the first instance, it would be impossible to completely isolate the effect of eccentricity using a spatial 2AFC test because the stimulus must be presented at some displacement left or right from the fixation point;

also, because the stimulus is moving, its eccentricity is constantly changing. Secondly, the experiment is more difficult for the subjects to perform accurately because it is an instinctive reflex to fixate upon and track moving objects, thus altering their effective angular velocity.

We therefore produced a best fit curve (Equation 5.6) which models the observed decline of temporal sensitivity more closely; and subsequently, we formulated a new model of spatiotemporal threshold for our application (Equation 5.8). It may be noted that this equation, as well as providing a more accurate and practical model, is also significantly less complex to compute than our original model.

5.2 User Study

So far we have considered the accuracy of our model in the task of predicting the visibility of various stimuli. This next study will undertake an analysis of the effect of our model on the user when it is used to reduce detail in a VE. This will be done by employing an objective analysis of the user's ability to perform a certain task, under normal and optimised conditions.

5.2.1 Overview

The measure which was elected to assess the user's performance was a wayfinding task, based upon the **simulated pursuit fixation** work of Cutting *et al.* (1992) and others (e.g. Warren and Hannon, 1990; Cutting, 1986; Rieger and Toet, 1985). This task involves the subject being passively transported through an environment of objects. The display is updated so that the observer is always looking towards a certain fixation point which is deviant from their heading direction. To illustrate this task in terms of a real world scenario, imagine that you are on the back of a jeep manning a TV camera which is free to pan left and right. The jeep is being driven through an environment cluttered with various objects, e.g. trees. You look through the camera's viewfinder and track one specific tree as the jeep moves through the environment, keeping the tree always in the centre of the viewfinder. Your gaze direction is now different from your heading direction. This concept is illustrated in Figure 5.10.

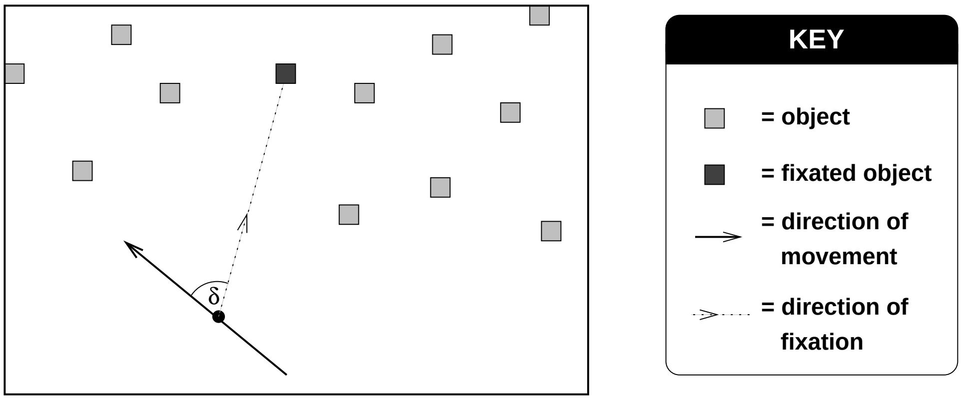


Figure 5.10: *Overview of the wayfinding task in which the subject is passively navigated through an environment with their direction of fixation oriented differently from their direction of movement. Angle δ represents the gaze/movement angle at one point on the navigated course.*

The task of the subject is to deduce whether they are being navigated to the left or to the right of the fixation point (based upon the radial motion cues which they acquire from the surrounding objects). For example, in Figure 5.10 the heading direction is to the left of the target. Intuitively, this task will become more difficult as the differential between the fixation and the heading vector (referred to as the **gaze/movement angle**) decreases. Typically, the gaze/movement angle will increase as one progresses along the navigated path. The maximum value of this angle for any trial is the independent variable of interest, i.e. the final gaze/movement angle.

From a number of subjects' responses to different navigation scenarios, Cutting *et al*. (1992) produced psychometric curves which plot the final gaze/movement angle against the percentage of correct responses. The point at which this curve drops below a certain threshold can be used to compare the subject's task performance under different situations. The reader should note that once again we are using a 2AFC paradigm (i.e. the subject is asked, 'did you move left or right?', as opposed to, 'did you move left?' or 'did you move right?').

This simulated pursuit fixation task was chosen as a measure of user performance because it offers a number of advantages which are particularly pertinent to our circumstances, e.g.

1. The subject must constantly fixate upon the centre of the display and so their physical gaze direction is always known. This therefore conveniently abates the requirement for a suitable eye tracking technology (which was not available to the author).

2. The environment is inherently motion-rich with many objects of different sizes and a good proportion of peripherally located features. It therefore offers good opportunity to exercise all aspects of our model.

3. This type of experiment, as well as having been used by numerous vision scientists in the past, has also been used by various researchers in the field of VR to assess the performance of subjects within a VE (e.g. Wann *et al.*, 1995). The technique is therefore an accepted and stable metric for assessing user performance.

5.2.2 Method

Stimuli. The test VE was generated using a Silicon Graphics, Inc. (SGI) Onyx RealityEngine^2 computer with one 200 MHz R4400 micro-processor, 128 MB of RAM and one Raster Manager (RM4) card with 4 MB of texture memory. The experiment was developed using IRIS Performer V2.0 for peak performance and used 'free running' phase mode (i.e. the application output a new frame as soon as it had completed the previous frame and was not constrained to a fixed frame rate).

The content and dimensions of the environment were modeled in order to replicate Cutting *et al.*'s (1992) experimental setup. Specifically, each environment contained 175 objects, randomly positioned at ground level (across the x–z plane), and randomly rotated (about the y axis). The fixation object was coloured light purple and a crosshair was positioned over it to guide the user's fixation.

Subjects were navigated through the environment for up to 5 seconds, but could submit their decisions at any point. The fixated object was initially positioned at a distance of 50 m from the viewpoint. Each object was 4.32 m high and the simulated forward velocity was 4.36 m/sec. The object which was used to populate the environment was the 'temple' model supplied by SGI. This was chosen because it was available in a convenient format and also because it contains a

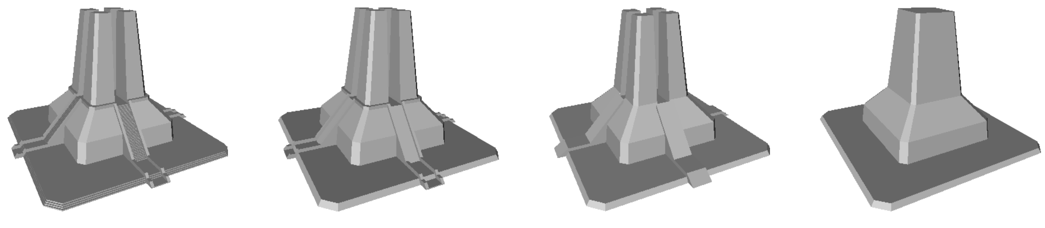


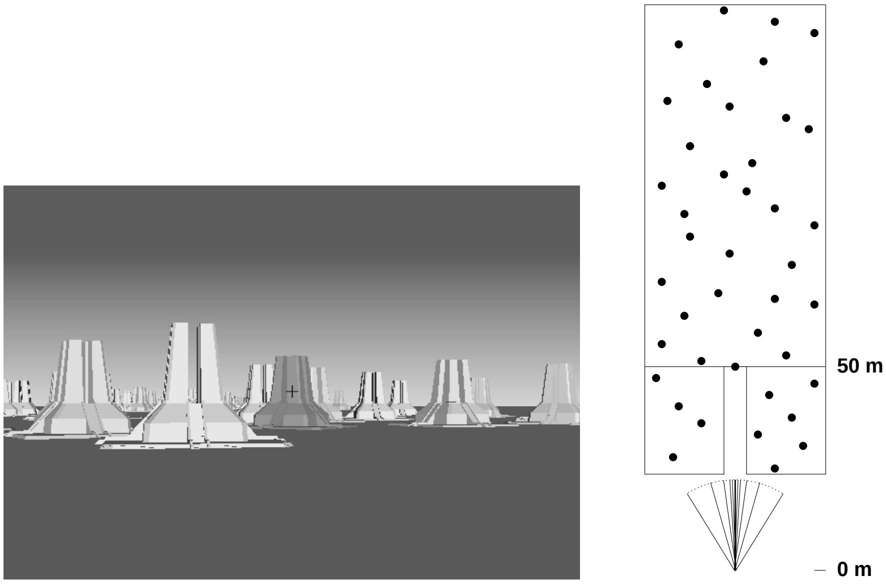
Figure 5.11: *The four levels of detail used for the object in the simulated pursuit fixation task. From left to right, these contain 3928, 834, 254, and 76 triangles respectively.*

broad range of spatial frequencies. Three further levels of detail for the test object were generated. These are illustrated in Figure 5.11 and a description of their spatial frequency content is provided in Appendix B. An overview and example screenshot of the test environment are provided in Figure 5.12.

Images were displayed at a resolution of 1280 $\times$ 1024 pixels. The subject viewed the screen such that it occupied 43.6 $\times$ 33.4 degrees of their field of view (and these FOV values were used by the graphics renderer for all perspective calculations). Steps were taken to ensure that no other users could remotely log into the workstation during the course of an experiment. This was done in order to limit the effect of any background processes which might interfere with the update rate of the experiment.

Sixty-four trials were randomly presented to each subject. These were composed of 8 final gaze/movement angles (0.25, 0.5, 1, 2, 4, 8, 16, and 32 degrees) $\times$ 2 gaze directions (left and right of the fixated object) $\times$ 2 LOD filtering conditions (with and without our perceptual optimisations) $\times$ 2 attempts. The model of threshold visibility which was used for the perceptual optimisations was based upon our new empirically-derived results (from earlier in this chapter), rather than the theoretical model we developed in Chapter 3.

Procedure. Twenty subjects participated in the study, drawn from a breadth of backgrounds including undergraduate students, postgraduate students, staff, and graduates of the University of Edinburgh. All subjects had normal or corrected-to-normal vision and were naïve to the experimental hypothesis. They were encouraged to keep their gaze fixated on the crosshair at the centre of the screen, but no attempt was made to monitor eye movements. A chin rest was used to restrict subjects' head movement and to maintain the viewing distance



(a) **(b)**

Figure 5.12: *An overview of the simulated pursuit fixation task. (a) presents an example screen shot of the test environment, and (b) provides a plan overview of the environment layout. The fan shape in (b) illustrates the various paths through the environment which were traversed. The fixation object can be noted at the centre of the 50m line.*

(which was set to 50 cm).

Subjects pressed either the left or right mouse button to indicate whether they felt that they had been transported to the left or to the right of the fixation point. Reaction times were recorded for each trial. No feedback was given to the subject on their success rate during the experiment to reduce the chances of them learning some nonsense visual task unrelated to wayfinding. Subjects were given a number of practice trials beforehand until they were satisfied that they understood the task. The experiment lasted about 20 minutes and participants were paid £5.00.

5.2.3 Results

Figure 5.13 presents the results of the simulated pursuit fixation experiment; averaged over the 20 subjects. These are compared for the case when our perceptual LOD optimisations are employed, and when no optimisations are used. On average, the act of implementing our optimisations produced a five-fold increase in frame rate (from 2.3 Hz to 11.5 Hz). It was anticipated that this performance increment would improve the subjects' ability to perform the heading task.

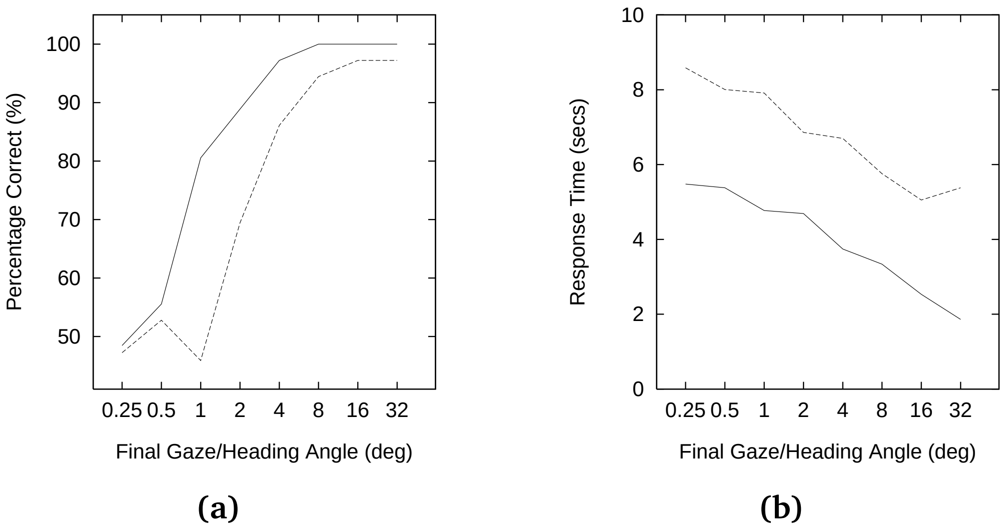


Figure 5.13: *Averaged results from the simulated pursuit fixation experiment. (a) shows the overall proficiency of subjects to perform the task. (b) presents the overall response times of subjects. In both graphs, the solid line represents the case where perceptual LOD optimisations were employed; whilst the broken line represents the case where no optimisation was performed.*

Figure 5.13(a) shows the average proficiency of subjects to correctly ascertain their heading. We can see from this graph that at large gaze/heading angles, the user is very proficient at resolving their heading direction; but this ability drops to chance level (∼50%) for smaller angles. It is immediately evident that subjects' efficiency was maintained higher for longer when our perceptual optimisations were employed.

Warren and Hannon (1990) used the interpolated final gaze/heading value corresponding to a 75% performance level in order to assess performance; whereas Cutting *et al*. (1992) proposed a more strict 95% criterion. We can therefore describe the subject's increase in performance in terms of these two definitions by referring to the data in Figure 5.13(a). Using Warren and Hannon's 75% threshold, subjects could discriminate final gaze/heading angles which were 3.0 times smaller (2.7 deg / 0.9 deg) using the perceptual LOD optimisations. Using Cutting *et al*.'s 95% threshold, subjects could discriminate angles which were 2.8 times smaller (9.5 deg / 3.4 deg) under optimised conditions.

Figure 5.13(b) shows the average response time of subjects during the task, i.e. how long it took them to resolve their heading direction. Again it is obvious from a cursory inspection that subjects had a distinctly faster response for cases when our perceptual LOD filtering was employed. From these data we find that, on average, users performed 1.67 times faster under optimised conditions.

5.2.4 Discussion

The findings of this task performance experiment show that for the prototype implementation of our perceptually-based LOD system (in comparison to the normal unoptimised case):

1. The frame rate of the simulation was increased (by around 5 times).
2. The subjects' ability to perform the heading task improved (they were able to resolve final gaze/heading angles up to 3 times smaller).
3. The time subjects took to perform each task decreased (they performed the task 1.67 times faster on average).

It is also worth noting that subjects did not report any visual discrepancies in the scenes which had perceptual LOD filtering applied, even when explicitly asked. It would therefore appear that we achieved our goal of reducing detail without affecting the user's percept of the VE.

5.3 Computational Studies

We summarised the goal of this doctoral thesis on Page 2 by stating that we would attempt to reduce lag in VEs by investigating perceptually-based LOD. We must therefore evaluate whether our system achieves this goal, i.e. to what degree does our prototype implementation reduce lag? An important issue which we should also address is that of the cost of implementing our perceptually-based system. Finally, it would also be interesting to investigate the extent to which each of the components of our model contribute towards any increase in performance.

All of the following computational studies used the simulated pursuit fixation experiment as their test application. These were run on the same platform as before (i.e. an Onyx RealityEngine^2 with one 200 MHz processor, 128 MB of RAM, and one RM4 card). All remote access to the machine was disabled for the duration of each study and all disk writes were made to the local disk, only upon completion of each trial.

All trials were performed under the same visual conditions as the previous task performance study, i.e. a display FOV of 43.6 $\times$ 33.4 degrees was assumed and the simulated forward velocity was 4.36 m/sec. All timings were recorded using a sub-microsecond resolution clock.

5.3.1 Analysis of Performance Speedup

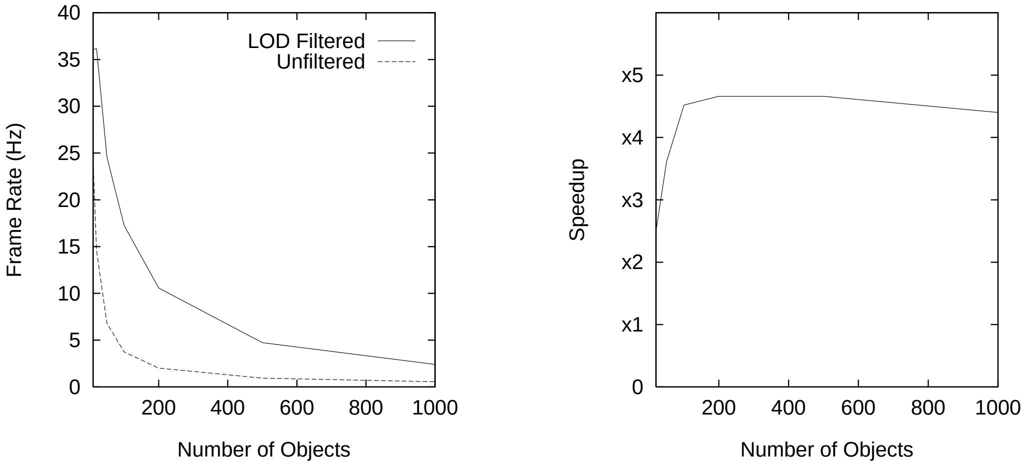
5.3.1.1 Objective

The aim of our first computational study was to assess the extent to which our prototype implementation reduces the latencies in a VE (i.e. the extent to which it improves performance). This was contrasted for a number of VEs with different numbers of objects in order to give an indication of how speedup varies with VE complexity for the test application.

5.3.1.2 Method

Procedure. The average frame rate (Hz) for a trial was found by dividing the number of frames rendered by the total time for the trial. Twenty trials were performed where the number of objects in the VE were kept constant (but their positions randomly varied). The final frame rate figure for such a set of trials was found by averaging the results of all 20 trials. Each set was repeated for the case were no LOD filtering was applied and when full perceptual LOD filtering was applied. Finally, a number of these sets of trials were performed for environments with different numbers of objects (10, 20, 50, 100, 200, 500, and 1000 objects). That is, 280 trials were performed in total (7 complexities $\times$ 2 LOD cases $\times$ 20 trials).

5.3.1.3 Results & Discussion



(a) **(b)**

Figure 5.14: *Results from the performance speedup analysis. (a) contrasts the average frame rate for the test application under normal unfiltered conditions (the broken line) and when perceptual LOD optimisations were employed (the solid line). (b) presents the data from (a) in terms of the relative speedup which was achieved when using the perceptual LOD optimisations over the normal unfiltered case.*

The results from this study are presented in Figure 5.14. From Figure 5.14(a) we can observe a consistent and marked increase in frame rate when our perceptual optimisations were employed. To describe the degree of this increment, Figure 5.14(b) illustrates the relative increase in performance which occurred when our perceptual optimisations were used. From this we can see that an average speedup of over $\times 4.5$ was quickly achieved (for environments with around 50 objects) and that this depreciated very gradually for more complex environments (> 500 objects).

Care should be taken when interpreting the initial sharp rise in speedup for environments with < 50 objects. This does not necessarily imply that less complex environments offer fundamentally smaller speedup values. We must also consider the fact that under the current implementation of IRIS Performer, the frame rate is constrained to be an integer multiple of the video refresh rate. Our video refresh rate was 72 Hz and so if the simulation was unable to run at 72 Hz, then it would drop down to the next integer multiple of 72, i.e. 36 Hz. Therefore, what we are most likely observing in the sharp speedup rise for < 50 objects is the point where the optimised environment is capable of being rendered faster than 36 Hz, but not as fast as 72 Hz, and so it is restricted to only 36 Hz. The relative speedup is therefore confounded by this additional factor for simple environments.

5.3.2 Analysis of Scheduler Complexity

5.3.2.1 Objective

Having investigated the performance gain that our prototype offers, we now wish to gain some insight into the cost of implementing our perceptually-based LOD model. That is, how much CPU time is spent evaluating the perceptual content and consequent optimal LOD for each object? Also, how does this change as the complexity of the VE changes?

5.3.2.2 Method

Procedure. Code was added to the prototype implementation to record the time before and after the LOD modulation routine. This was taken as a pragmatic measure of the time spent implementing our perceptual model for each frame. The final figure for any one trial was found by averaging the times for every frame of the simulation. Twenty trials were performed where the number of objects in the VE were kept constant (but their positions randomly varied). The final figure for such a set of trials was found by averaging the results of all trials. A number of sets of trials were performed in order to find the average time complexity for environments with different numbers of objects (10, 20, 50, 100, 200, 500, and 1000 objects). That is, 140 trials were performed in total (7 complexities $\times$ 20 trials).

5.3.2.3 Results & Discussion

Figure 5.15 presents the results from the analysis of our prototype scheduler. We can consequently observe that the scheduler exhibits an essentially linear increase in processing requirements as the VE becomes more complex. For the host architecture which was used, these data imply that our scheduler would not begin to interfere with a 30 Hz frame rate until around 450 objects were introduced (assuming that no other intensive processing is performed per frame),

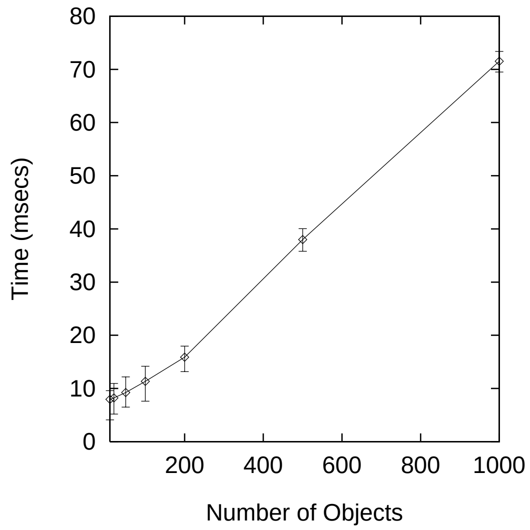


Figure 5.15: *The overhead of implementing our perceptually-based LOD system for a range of VE complexities. This includes the spatial frequency estimation as well as the calculation of object visibility and the subsequent LOD selection.*

and it would take over 900 objects to begin to interfere with a 15 Hz frame rate.

It is worth noting that when we say 900 objects, this does not necessarily mean a VE with only 900 objects. We are of course referring to *degradable* objects here (i.e. objects which have multiple levels of detail) and not all objects in a VE will be degradable. The number of non-degradable objects in a VE will have no effect on the computational resources consumed by the scheduler because these objects should never be considered by the scheduler.

It is also worth noting that these figures are for our prototype implementation. Being a prototype, the author was more concerned with the functionality of the implementation rather than its efficiency. It is therefore highly likely that a more efficient implementation could be developed. (For example, the results from our model of threshold visibility could be placed into a look up table rather than being computed directly each frame. Also, the author used the IrisGL character string routines to calculate the projected screen coordinates for each object. These return screen coordinates as a side-effect but also perform a number of other functions which are unrelated to our task. A more efficient implementation would therefore calculate these projected coordinates directly.)

5.3.3 Breakdown of Model Components

5.3.3.1 Overview

In this final computational study we will attempt to gauge the extent to which each of the three principal components of our model are utilised. That is, how much does size LOD, eccentricity LOD, and velocity LOD contribute towards the total increase in performance for our test application.

It would be an unfair assessment to analyse the velocity and eccentricity components of the model in isolation because most of the benefit of these two components is realised through their synergic combination (we will show this in the next chapter). Therefore we shall compare the case where all optimisations are used, against the case where only the velocity and eccentricity based optimisations are used. In effect, this provides us with an indication towards the additional benefit which can be accrued over traditional size LOD when we also incorporate optimisations based upon the velocity and eccentricity of objects.

5.3.3.2 Method

Procedure. The average level of detail (1–4) for a frame was found by dividing the sum of all objects' LOD by the number of objects. An average figure for an entire trial was found by dividing all of the frame averages by the total number of frames displayed. As before, 20 such trials were performed, and the average LOD figure from all of these was found. This measure was used as an assessment of the degree to which detail was optimised; with higher values indicating a greater degree of optimisation.

Two level of detail cases were compared: one where full optimisations were employed (size, velocity, and eccentricity), and the other where detail was reduced based only upon the velocity and eccentricity of objects. This relationship was analysed for environments of various complexities (10, 20, 50, 100, 200, 500, and 1000 objects). 280 trials were therefore performed in total (7 complexities $\times$ 2 LOD cases $\times$ 20 trials).

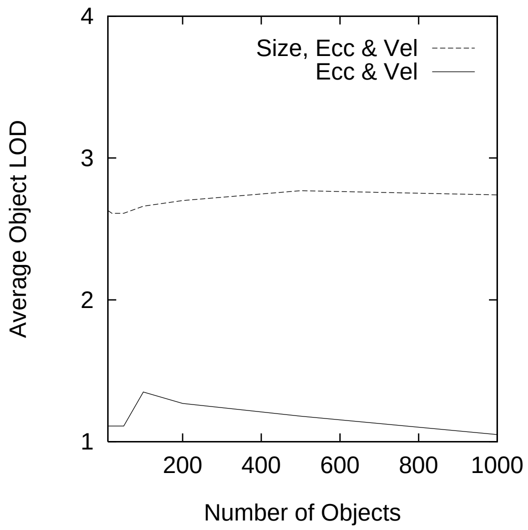


Figure 5.16: *Comparison of average LOD during a trial with and without size optimisations. (Recall that the four levels of detail for each object are illustrated in Figure 5.11.)*

5.3.3.3 Results & Discussion

Figure 5.16 presents the data from this study. The most obvious result from this graph is that size optimisations appear to account for the bulk of any reduction of detail (and hence the increase in performance). The combined contribution of velocity and eccentricity optimisations rose to a maximum of 21% of the total LOD reduction for around 100 objects. This dropped down to 10% for 500 objects, and to as little as 3% of the total reduction for 1000 objects[1].

Of course, this analysis was performed for environments with a fixed FOV and (relatively slow) forward velocity. If we were to increase the forward velocity through the environment then the velocity component of our model would be given more chance to contribute towards the overall reduction in detail and performance should improve as a result.

It would be tempting to make the same analogy for FOV, but this would be a specious assumption. Although increasing the FOV of the display would allow objects to exist at greater eccentricities (and hence theoretically allow them to have their detail reduced further), we must also consider the angular resolution of the display. If we take the same display and simply stretch it over a wider

[1] This drop in the contribution of the velocity and eccentricity components might explain the gradual decline in speedup which was found for environments with > 500 objects, as shown in Figure 5.14(b).

field of view, then the individual pixels will become larger and so the angular resolution—and hence the highest visible spatial frequency—will drop. So, although the field of view is larger, objects must move further into the periphery before their detail drops below threshold and a lower LOD can be selected[2].

5.4 Summary

We have now completed the evaluation of our visibility model and prototype implementation, and can therefore assess the implications of our findings. (N.B. a deeper discussion of the various implications of these results will be performed in the next chapter.)

We began by investigating the accuracy of our model; employing various psychophysical studies to compare the perceived threshold of different stimuli with the predicted thresholds from our model. The resulting data showed that our model predicted the size threshold for objects with sufficient accuracy. The predicted eccentricity thresholds were very close to those found through experiment; however we produced a new equation (Equation 5.5) to better model the empirical data. In the case for velocity thresholds, we found quite a large discrepancy between the predicted and observed response. This was accounted for and a new, more simple model was devised (Equation 5.8) based upon the observed results.

Our formulation of these new models of visual acuity is of course valid because we are concerned with sufficiently different stimuli from those which normally preoccupy vision scientists. For example, we are concerned with local, aperiodic, non-harmonic stimuli whilst vision scientists normally deal with contrast gratings which are harmonic, periodic, and extend over a large field of view. We have therefore tailored our model to the genus of stimuli commonly found in computer-generated imagery.

The second section of this chapter presented a comparative analysis of a user's ability to perform a prescribed task. For the particular simulated pursuit fixation

[2]This problem poses a common dilemma in the field of head-mounted display design, where manufacturers must decide whether they wish to produce an HMD with high angular resolution or wide FOV, because the pixel resolution of current LCD technology is limited.

task chosen, we found that the act of implementing our perceptual LOD model improved the frame rate of the simulation markedly. As a result, the user's ability to accurately perform the task improved, as did the speed with which they could complete the task. All of this was achieved with no reported drop in image fidelity. These results vividly illustrate the merits of this research effort.

The final investigation focussed on the basic computational benefits which resulted from the implementation of our system. These studies were performed using the simulated pursuit fixation experiment and so the results pertain only to that particular application. However, this application is not completely artifical; for example, the behaviour and operation of the experiment are very similar to that of a driving simulator. The author therefore feels that these results provide a good representative indication of the general utility and performance which can be offered by perceptually-based LOD. To reinforce this, it is worth noting that various other researchers have achieved comparable performance increments using size and eccentricity LOD, e.g. Ohshima *et al*. (1996) experienced a five-fold improvement in frame rate for their LOD system, and Levoy and Whitaker (1990) experienced an improvement in rendering performance of $\times$ 4.6 for their perceptually optimised ray tracer.

We found that the frame rate of the simulation was improved dramatically by our system (around four to five-fold speedup), and that this improvement was maintained for VEs over a large range of complexities. We also saw that the implementation of our system can be made very efficient, such that it would not affect the update rate of the simulation until an excessively large number of objects is introduced (in the order of 1000 degradable objects). Finally, we found that out of the three major components of our model (size, velocity, and eccentricity), the accommodation of size-based optimisations provided the largest contribution towards the subsequent improvement in performance.

Chapter 6

Discussion

‘ “Begin at the beginning,” the King said, gravely,
“and go on till you come to the end: then stop.” ’

(Lewis Carroll, *Alice in Wonderland*)

This chapter provides a focal point to discuss many of the issues pertaining to the subject matter of this dissertation. This will be done from several perspectives. We will begin by examining the specific issues associated with our particular model and its implementation. Following this, we will discuss the more general matters relating to any computer graphics system which attempts to modulate detail using perceptual criteria. Finally, we will offer some comment on the wider issues of applying knowledge from the domain of visual perception to that of computer graphics.

6.1 Discussion of our Model and Implementation

This first section deals with the specific model and implementation that we have developed as a result of this research effort. Through this examination we will review how successful the work has been, revise our visual acuity model based upon our experimental results, contrast a few of the major design decisions that were taken, and finally, we will consider the extension of our model to support fixed frame rate LOD.

6.1.1 Reviewing the Results from our Prototype Evaluation

The results from the previous chapter offer a proof of concept for the notion of perceptually modulated LOD in general, and our implementation in particular. They show quite plainly that such a system can substantially improve the performance of a VE, resulting in increased user performance; and that this can be done with no detrimental effect on the perceived fidelity of the VE. These results provide an indication of the amount of visual detail that a standard graphics renderer produces which the user may never see.

During the Introduction chapter, we listed a number of problems with the typical implementation of LOD and stated that we would address these in this thesis (see Section 1.3.3, page 11). Let us therefore review the success of our work in terms of these four criteria:

1. **Principled Selection Mechanism** : our model has been developed with close reference to contemporary theories and models of visual perception. This has resulted in a principled mechanism to select level of detail. There is no need for the user to provide any subjective input, or to use any trial and error judgements to find the best LOD switching conditions.

2. **Eliminate Popping Effects** : our model attempts to locate the optimal LOD to use at any point, i.e. the lowest model such that the user cannot perceive any difference between this and the highest LOD. As a result, the user will not experience any noticeable or distracting anomalies when different LODs are selected. The results from the previous chapter confirm this achievement.

3. **General LOD Framework** : our model is general, in that it encompasses all of the principal factors which can affect an object's visibility (i.e. size, eccentricity, and velocity). It is also orthogonal, in that any combination of these factors can be used selectively without compromising the accuracy or function of those parts of the model.

4. **LOD Generation** : we acknowledge that LOD generation techniques are integral to the topic of LOD. We have therefore developed a perceptually driven polygon simplification framework to complement the implementation of a perceptually modulated LOD system. This attempts to ensure that the extent of a simplification is restricted to a prescribed spatial threshold.

6.1.2 Re-implementing our Visibility Model

Using the results from our eccentricity and velocity psychophysical studies, we can now present a re-implementation of our visibility model, i.e. the phase that calculates α_{vis} for an object. Previously, we had to employ an iterative process in order to locate a user's threshold spatial frequency for a moving stimulus (see Section 4.4.4.1, page 123). However, using our empirically derived formula from Equation 5.8 we can now compute this value directly. The resulting frequency can then be scaled by our revised definition for cortical magnification (Equation 5.5) to produce a value for the highest visible spatial frequency at any eccentricity and velocity. This is described by the following formula for visual acuity (c/deg):

$$H(v, E) = G(v) \times M, \tag{6.1}$$

where,

$$G(v) = \begin{cases} 60.0, & \text{when } v \leq 0.825 \\ -27.78 \log_{10}(v) + 57.69, & \text{when } v > 0.825 \end{cases} \tag{6.2}$$

$$M = \begin{cases} 1.0, & \text{when } E \leq 5.79 \\ 7.49/(0.3E + 1)^2, & \text{when } E > 5.79. \end{cases} \tag{6.3}$$

Figure 6.1 presents a 3D graph of the function $H(v, E)$. This spatiotemporal threshold surface shows the visual acuity of a standard observer at any point in their peripheral field and at any angular velocity. This surface has been specifically tailored (by a process of empirical evaluation) to the genus of complex stimuli that occur within computer-generated imagery; rather than the simple, harmonic stimuli used by vision scientists.

The surface provides a description of the general ability of a user to perceive detail. We can apply this to a particular display device by thresholding the spatiotemporal surface with the appropriate value of highest displayable spatial frequency (ξ). To illustrate this process more clearly, the following source code segment presents a simple C++ implementation of the entire phase to calculate the value of α_{vis} for an object.

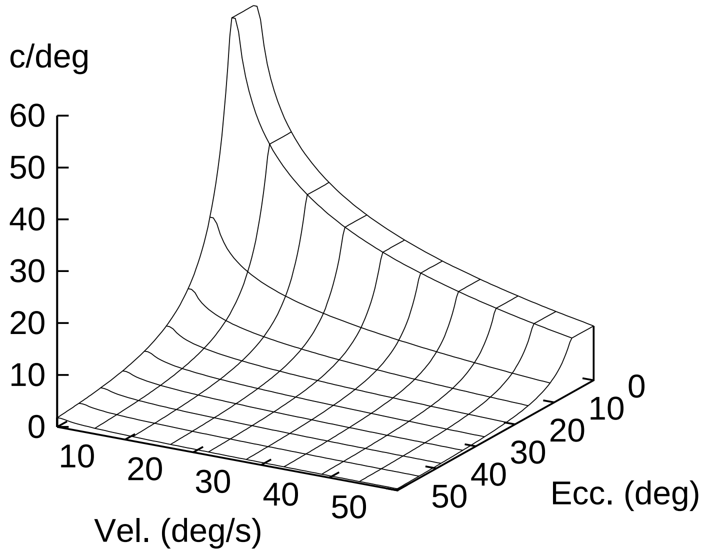


Figure 6.1: *The spatiotemporal threshold surface defined by our new visibility model (Equation 6.1).*

```
// Given values for eccentricity (deg), velocity (deg/s), and
// the display device's highest displayable frequency (c/deg),
// return the highest visible spatial frequency (c/deg).

float spatFreqLimit( float ecc, float vel, float maxDispSF )
{
   float motionSF = 60.0, cortMag = 1.0, result, temp;

   if ( vel > 0.825 )
      motionSF = -27.78 * log10(vel) + 57.69;

   if ( ecc > 5.79 ) {
      temp = ( 0.3 * ecc ) + 1.0;
      cortMag = 7.49 / ( temp * temp );
   }

   result = motionSF * cortMag;
   return ( result < maxDispSF ) ? result : maxDispSF;
}
```

This computation is of course only performed once for each degradable object during a frame. However, if we wished to optimise this routine in order to avoid the potentially expensive log operation and floating point arithmetic, then we could precompute the results of this function beforehand and store these in a look up table.

6.1.3 Contrasting our Design Decisions

Now that we have looked at the details surrounding our particular model, it may be worth discussing some of the major design decisions that were taken and contrast these against any alternative approaches which could have been adopted. This should also help to place our work into context with similar components of other systems.

6.1.3.1 Image versus Geometry for Measuring Detail

The pivotal design decision of our model is that we assess detail within a scene based upon the rendered image of an object, rather than its geometry. This is not a novel approach in itself (e.g. Maciel and Shirley, 1995, used an arbitrary image based metric to assess whether the modulation between two object representations might be perceivable from a certain viewpoint). However, there are a number of implications arising from this decision.

We have already examined many of the pros and cons for both the image based and geometry based approaches (see Section 3.1.1, page 67). We will therefore not replicate that discussion here. Suffice to say that any system which purports to utilise models of visual perception, must avail itself of the actual information which an observer perceives. For example, a purely geometry based approach could not describe the perceived effect of a texture mapped polygon. Also, a collection of adjacent polygons may be perceived as a single visual feature; however a geometry approach would simply use the sizes of the component polygons to decide the visibility threshold and hence underestimate the perceived size of that region.

The major limitation of our image based approach is that it cannot account for changes in environmental conditions (e.g. lighting, fog, etc.) because the spatial frequencies are collected off-line. However, it is worth noting that a geometry based approach would most probably suffer from this same limitation. We therefore conclude that an image based system is the correct approach to adopt for our particular application. Unfortunately, a complete model of perceptual image segmentation does not exist, so we were forced to develop an interim solution based upon colour difference theory. Given the results from the previous chapter, it would appear that this offers an adequate solution.

6.1.3.2 Traditional versus Hierarchical LOD

Most LOD systems operate by storing a collection of independent models for an object and switching between these representations during the simulation. An alternative method is to contain all of the various levels of detail for an object within a single data structure. For example, a quad tree structure can be used such that the deeper you progress through the tree, the more detail is added to the model. This technique is often used to provide continuous levels of detail for terrain databases (e.g. Lindstrom *et al.*, 1996; Economy *et al.*, 1990).

One advantage of hierarchical LOD over traditional techniques is that different parts of an object could be displayed at different levels of detail. From our perspective this would mean that if a large object is displayed on screen, then any parts of the object that are within the user's peripheral field could potentially be degraded whilst retaining high detail for those parts of the object which the user is focussing upon (Sen *et al.*, 1995). This would be most useful in the situation where the VE contains a small number of large objects: a situation in which traditionally implemented eccentricity LOD would not cope well.

There are however certain logistical difficulties associated with hierarchical LOD. Firstly, there may be a large development effort required to implement this technique: not all image generators may provide the ability to easily implement this; whereas most, if not all, renderers support a mechanism to implement traditional LOD (Reddy, 1995b). There is also the question of how to hierarchically decompose an arbitrary polygon mesh, and to do so in such a manner that there are no discontinuities when different parts of the model are displayed at a different detail level. This is a more simple problem for terrain databases because these data are composed of a uniform grid of height fields.

We have decided to adopt the more traditional approach to LOD because of its conceptual and computational simplicity, and also because of its generic applicability to any graphics system. This should make our work accessible to a wider audience, and in particular to users on low end machines.

6.1.3.3 Empirical versus Simulated Contrast Sensitivity

Sen *et al.* (1995) have proposed a model of the human retina in order to assess the contrast sensitivity of a user. This involves applying physiological data on the distribution and receptive field size of neurons in the retina, and then simplifying these to produce a manageable system. This model is then used in order to derive the expected contrast sensitivity for any region of the visual hemisphere.

This represents a fundamentally disparate approach to the one that we have taken, i.e. our contrast sensitivity model is based upon empirical studies of various observers' ability to resolve actual stimuli; whereas Sen *et al*.'s approach is to simulate users' perception using a model of the retina's structure.

There are a number of implied assumptions associated with Sen *et al*.'s approach which need to be resolved. For example, although it is clear that retinal ganglion cells have a bounding effect on the stimuli that we perceive, there are many other stages to perception which also affect the visual data, e.g. the retinal model approach discounts all processing that occurs in the visual pathways and visual cortex.

Having said this, the work of Sen *et al*. is most intriguing, and the author looks forward to discovering how their work progresses. It would be particularly interesting to compare the results from their simulated contrast sensitivity model with that of our empirically derived model.

6.1.4 Extending the Model for Fixed Frame Rate LOD

During our literature review of LOD techniques in Chapter 2, we stated that fixed frame rate LOD (degrading each object's LOD in order to achieve a desired constant update rate) is not directly relevant to our work (Page 38). However, we can suggest a way in which the model that we have developed may be extended to support fixed frame rate LOD in a perceptually principled manner.

If we were to add a global scaling factor, $s \in [0..1]$, to the computation of highest visible spatial frequency (α_{vis}) for all objects, then this would allow us to dynamically vary the magnitude of a user's threshold vision based upon the current system stress. When s is given the value 1.0, then the LOD system will act in the manner that we have already described; however if we choose a lower value of s then the system effectively believes that the user's visibility threshold is lower, and hence it will attempt to reduce the detail of objects that are of a larger size, a lower eccentricity, and a slower velocity.

For example, we could use this approach to produce a simple reactive fixed frame rate system. In this case, if the time for any frame exceeds the desired frame period, then we simply reduce the value of s accordingly. Alternatively, if we find that we have spare compute time after completing a frame, then we can increase the value of s until it reaches 1.0 (there is of course no need to exceed a scaling factor of 1.0). It is important to note that this enables us to develop a fixed frame rate system that will degrade resources in a perceptually linear manner, i.e. the less overloaded the system is (i.e. larger s), then the less perceivable the changes in scene complexity will be, and vice versa.

6.2 Discussion of Perceptually Modulated LOD

Up to this point we have discussed and contrasted various features of the specific model proffered by this thesis. However, there are a number of general considerations which relate to any system that implements perceptually modulated LOD. This section will cover a few of these issues and also undertake some examples to illustrate the potential merits of perceptually modulated LOD.

6.2.1 Considerations for Perceptually Modulated LOD

The following list presents various factors that apply to the general topic of optimising visual detail in a computer graphics system using perceptual criteria. These points are therefore applicable to the system that we have developed, as well as any future systems which may appear.

6.2.1.1 Desktop versus Immersive

A true perceptually-based LOD system should incorporate eye tracking, or at least head tracking, to know where the user is gazing. This means that a VE is optimised based upon the perspective of a single user, i.e. the system can only support one user per display device. This would imply that perceptually modulated LOD would be more suited to an immersive VR system where this situation is more generally true.

6.2.1.2 Rendering Consideration

Some researchers have contemplated using different rendering models to produce various LODs, such as flat-shading or smooth-shading (e.g. Funkhouser and Séquin, 1993). However we should note the existence of a fundamental dichotomy in this case. That is, a smooth-shaded object will generally have a lower spatial frequency content than the corresponding flat-shaded object (because the interpolation removes sharp edges from the image). However, smooth-shading is a more computationally intensive operation than flat-shading (because the surface normal is found for every pixel rather than for the entire surface). We therefore have the conflicting situation where smooth-shading is perceptually less complex, but computationally more expensive (see Figure B.4, Page 202). We therefore suggest that different rendering models should not be used to produce different levels of detail.

Another rendering consideration worth noting is that of antialiasing. The use of antialiasing techniques can improve the perceptual accuracy of an image. For example, they can be used to prevent the scintillating effect which is caused when near sub-pixel features continually appear and disappear from an image. We therefore advocate the use of antialiasing where this does not compromise the performance of a VE (e.g. if it is implemented in hardware).

6.2.1.3 Simplification Considerations

We made the recommendation in Section 4.3.1 (Page 111) that levels of detail for an object should be produced using a simplification algorithm which incorporates some means to limit the spatial extent of a degeneration. We should highlight here that this is not a specific requirement of our model, but a general requirement for perceptually-based LOD. In order to modulate detail using perceptual criteria, we require models with a range of visual as well as geometrical complexities.

One could of course use a standard polygon simplification algorithm with no framework for limiting the visual effects of the degeneration. However this may produce models which do not lend themselves well to selection based upon perceptual criteria; and as a result, the system would be unable to optimise the display as efficiently as might otherwise be possible.

6.2.1.4 Modeling Issues

We are not directly concerned with modeling issues in this thesis (beyond the generation of various LODs). However, we will attempt to bridge two important questions: how many levels of detail should be generated for an object, and at which complexities? These are notoriously difficult questions to answer because they will depend heavily upon the specific platform, the graphics renderer, the complexity and topology of the object, and the application in question. It is highly unlikely that a suitable metric could be developed to predict, *a priori*, the best suite of LODs to use in any situation.

However, we can suggest that any attempt towards a solution would be best achieved using some form of feedback loop; where the LODs for an application are adaptively refined through a pragmatic benchmarking process. We note that this might involve two distinct stages of feedback to the LOD generation phase, in order to improve the perceptual and the computational efficiency of the LOD suite, respectively. For example:

1. Firstly, we wish to ensure that we have LODs over a wide perceptual range; so that there is more chance that an object can be degraded under low visibility conditions. This could be done by comparing the spatial frequency profile of each LOD to see if they are suitably different. (N.B. our LOD generation framework is particularly suited to producing models with a range of perceptual profiles because we can vary the scale at which a reduction is performed.)

2. Secondly, based upon the usage of the LODs in a typical invocation of the application, we can find which models are used the most frequently and attempt to increase the range of LODs around this value. Also, if certain models are never used then we can alter the range of perceptual complexities accordingly.

6.2.2 Assessing the Value of Perceptual Optimisations

Throughout this thesis we have referred to the reduction of detail in an object contingent upon its eccentricity and velocity across a display. However, we have given little consideration to the physical merits of this process, i.e. how fast does an object have to travel, or how far does it have to progress into the periphery, before we can reduce its detail? And are these situations likely to occur regularly as a user navigates through a VE? These questions will be dealt with now.

6.2.2.1 A Desktop Example

To begin, let us take the example of a typical desktop display: 40×30 cm screen size, 1280×1024 pixel resolution, and viewed at a distance of 40 cm. Using Equations 3.2 and 3.3 (Page 77), we can calculate the display's field of view as: 53.1×41.1 deg. Finally, we can use Equation 4.4 to find the highest angular resolution of the display in terms of spatial frequency: 12.0×12.5 c/deg.

We can therefore see that for this example we can never be presented with a stimulus of a spatial frequency greater than 12.5 c/deg. This is about one fifth of the maximum spatial frequency that we can generally resolve (60 c/deg). So what effect does this have on our system?

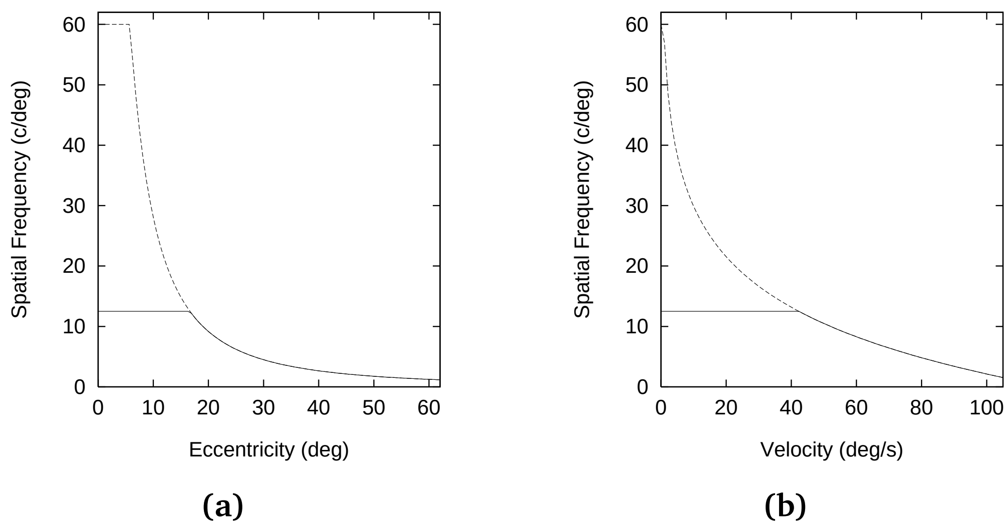


Figure 6.2: *The highest visible spatial frequencies for a user viewing a display where ξ = 12.5 c/deg. These are shown for (a) increasing eccentricity, and (b) increasing velocity. The eccentricity curve (broken line, a) is defined by Equation 5.5, and the velocity curve (broken line, b) is defined by Equation 5.8.*

Figure 6.2 presents graphs of our new models for cortical magnification and velocity. Both of these have been thresholded using a highest displayable spatial frequency of 12.5 c/deg to illustrate the potential stimuli in our desktop example. From these two graphs we can observe firstly that we do not begin to perceive less detail in our peripheral field until an eccentricity of $\sim$17 deg is reached. More striking than this is the result that our effective spatial perception will not degrade until an object exceeds a velocity of $\sim$42 deg/s.

In terms of our display example (which, recall, occupies 53.1×41.1 deg of arc) these results mean that an object would have to be displaced horizontally from the focus point by around one third of the display—or that it would have to travel from the left edge of the display to the right edge in around 1.26 seconds—before the system could even contemplate reducing the detail of an object.

Of course, these are extreme and isolated cases. When we combine the effect of both velocity and eccentricity, then the above figures will be substantially mitigated. For example, at an eccentricity of 10 deg, then an object need only travel above $\sim$13 deg/s before detail could become potentially invisible. It is therefore evident that, taken on their own, eccentricity and velocity based optimisations will probably provide a meager performance advantage; however, if the two are implemented together, then this will produce a synergic speedup.

6.2.2.2 An Immersive Example

We have already suggested that perceptually modulated LOD might be better suited to an immersive VR system rather than a desktop system. Let us therefore investigate the implications of such systems with reference to the previous desktop example.

Table 6.1 provides a cross-section of modern head-mounted displays. For each entry we have calculated the corresponding highest displayable spatial frequencies for that device[1]. These figures should be compared with the 12.0×12.5 c/deg resolution from our desktop example.

Let us examine further the affordances of the i-glasses! HMD. This LCD based unit is indicative of currently popular, cheap HMD systems; both in terms of resolution and FOV. From a casual inspection we can see that this device has a significantly lower angular resolution than our desktop example, almost three

[1] It is interesting to note an obvious technology divide emerging from Table 6.1: the three highest resolution units (MedView, Datavisor, and SIM-EYE) are all based on Cathode Ray Tube (CRT) technology; whereas all the other devices are Liquid Crystal Displays (LCDs).

HMD Model	Field of View	Pixel Resolution	Highest Displ. Spatial Freqs.
CE-100S	22.5 × 16.8 deg	208 × 155	4.6 × 4.6 c/deg
CyberFace 2	140 × 110 deg	385 × 119	1.4 × 0.5 c/deg
CyberMaxx	58 × 42 deg	267 × 225	2.3 × 2.7 c/deg
dVISOR	105 × 41 deg	294 × 141	1.4 × 1.7 c/deg
Datavisor	50 × 37 deg	1280 × 1024	12.8 × 13.8 c/deg
EyeGen3	32 × 24 deg	332 × 493	5.2 × 10.3 c/deg
i-glasses!	30 × 24 deg	263 × 230	4.4 × 4.8 c/deg
MedView	30 × 30 deg	1280 × 960	21.3 × 16.0 c/deg
MRG 2.2	84 × 65 deg	240 × 240	1.4 × 1.8 c/deg
MRG 4	61 × 46 deg	160 × 234	1.3 × 2.5 c/deg
SIM-EYE 60	100 × 60 deg	1280 × 1024	6.4 × 8.5 c/deg
VIM 1000pv	100 × 30 deg	710 × 225	3.6 × 3.8 c/deg
VR4	48 × 36 deg	247 × 230	2.6 × 3.2 c/deg

Table 6.1: *A comparison of the highest displayable spatial frequencies for a number of contemporary HMDs. The figures for field of view and pixel resolution are taken from VR News (1995). The corresponding figures for highest displayable spatial frequency were calculated using Equation 4.4.*

times so (compare 4.4 × 4.8 c/deg with 12.0 × 12.5 c/deg); also, the field of view is considerably smaller than the desktop case (30 × 24 deg versus 53 × 41 deg).

If we repeat the same analyses that we undertook for the desktop example, then we find that for $\xi = 4.8$ c/deg, an eccentricity of 29 deg, or a velocity of 80 deg/s, must be exceeded before it would be possible to degrade objects without the user being able to perceive the modulation (refer to Figure 6.2). As the horizontal FOV for the device is 30 deg, it is therefore apparent that employing only eccentricity LOD will have essentially no benefit. Also, when employing only velocity LOD, then an object must travel across the entire display in under 0.375 seconds before any optimisation can occur. Once again however, if we take these two components together, then the situation is less drastic: e.g. at an eccentricity of 15 deg, then an object need only travel at 24 deg/s before we may begin to reduce LOD. The exact relationship between eccentricity and velocity for this example can be more easily ascertained by examining the surface in Figure 6.3.

There is a further benefit in immersive VR systems because, whenever the user

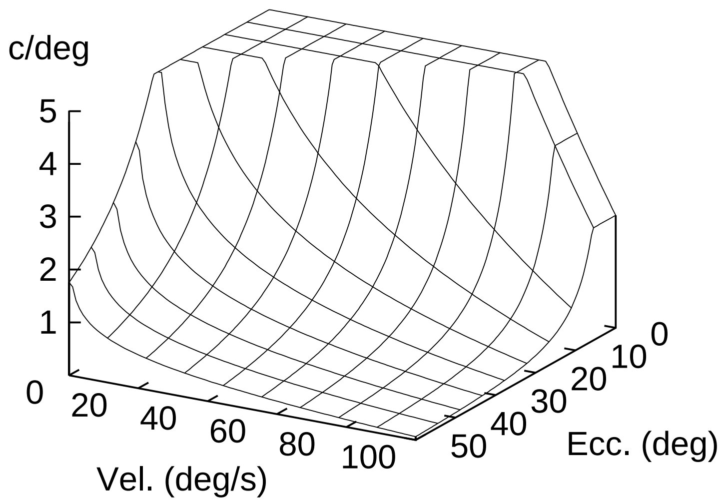


Figure 6.3: *The effective spatiotemporal threshold surface for the i-glasses! HMD, i.e. this represents the graph from Figure 6.1, thresholded with a highest displayable spatial frequency, ξ = 4.8 c/deg.*

moves their head, the whole world effectively moves (in relation to the user's viewpoint). A user's head rotation can often be in excess of around 180 deg/s; thus providing considerable opportunity for most objects to be degraded during exaggerated head movements. This is particularly desirable because many users tend to feel disoriented when they move their heads rapidly and the system does not keep up with their movements (Holloway, 1991). Therefore, by reducing detail during head movements we can improve the update rate of the system and thus make the simulation appear more smooth and interactive. Our model will of course automatically support this feature because we measure the velocity of objects across the display device, i.e. relative to the user's viewpoint.

In summary then, for many of the current generation of LCD HMDs, their lower angular resolution and smaller FOV means that there would be practically no benefit in simply using either eccentricity or velocity optimisations on their own (that is, if the requirement is to be maintained that the user should not be able to perceive the modulation of detail). However, by using the two optimisations in parallel, their contribution can become significant (refer to Figure 6.3). This will be particularly evident in situations when the user makes large head rotations. This latter optimisation could prove a major factor in making immersive VR systems appear less disorienting.

6.3 Discussion of General Perceptual Issues

The following sections will attempt to take a step back and reflect upon the more general issues of applying knowledge from the domain of visual perception to that of computer graphics. This includes fostering an appreciation for the subjective nature of many visual theories, and in particular for the principles upon which this thesis is founded. We will also delve into a number of relevant or interesting topics within the field of visual perception and illustrate how these may apply to our work.

6.3.1 Acknowledging the Extent of Perceptual Knowledge

As the reader has progressed through this thesis, they may have developed a feeling for the incompleteness of our understanding of visual perception. This is certainly true: there is no grand unified theory of perception that conveniently explains all perceived phenomena. This is not a criticism on the part of vision scientists. On the contrary, the discoveries and advancements which have been made in this field are estimable given the sheer complexity of the brain which must be fathomed. To illustrate this, the author Lyall Watson once made the perspicacious comment,

> '*If the brain were so simple that we could understand it, we would be so simple that we couldn't understand it.*'

Even Albert Einstein conceded,

> '*How difficult it is! How much more difficult psychology is than physics.*'

The difficulty of visual perception is beautifully encapsulated by Marr (1982) who develops the analogy of a computer system. Essentially, if we were simply to look at the fluctuation of ones and zeros in a computer memory, could we really decipher the underlying structure and design of a modern computer? This is a commensurate problem to that facing vision scientists who must try to understand perception by only analysing the state of millions of neurons in the retinae and visual cortex.

It is therefore important to appreciate that there are limits to our perceptual knowledge. For example, in the context of this thesis, we know about the contrast sensitivity of the eye, and how this can be measured and modeled; however we do not know how the brain segments the retinal image into component features so that we may accurately apply our contrast sensitivity models to predict the visibility of an arbitrary scene.

Additionally, even if we do understand the underlying principles for a small area of visual perception, we must also appreciate that our visual system is an extremely adaptive and tolerant machine. There will always be a degree of subjectivity when we talk about perception because this faculty is, by definition, a personal perspective and will vary marginally for different people. The following section will illustrate this latter point with specific reference to the subject matter of this thesis.

6.3.2 A Note on Perceptual Subjectivity

We made the point in Section 2.3.4.1 (Page 64) that each individual's visual system will exhibit subtle nuances, and that our contrast sensitivity model therefore only pertains to a standard, averaged observer. In general, the notion of a standard observer is valid because we find that most individuals tend to have a visual performance close to this ideal (the reader is directed to Figures 5.5 and 5.8 for evidence of this).

However, we will take this opportunity to present a list of factors which can affect a user's visual perception, and thus specify the various variables which can introduce a degree of subjectivity into our perceptual algorithms. We can classify these into two distinct classes: environmental (to do with the state of the environment which the user occupies), and individual (to do with the state of the user's specific visual system).

Environmental Considerations :

- **Background Illumination** : the background light intensity which is used to illuminate a stimulus can affect its visibility. Kelly (1975) presents theoretically derived contrast sensitivity functions for an observer under a range of background illuminations. These show a degradation of sensitivity in dim lighting conditions. Following from Lamming's (1991a) exposition, we can attempt to explain this effect from a quantum perspective: under dim conditions, there will be less light quanta entering your eyes and so less opportunity that sufficient information can be gathered to form a detailed image within the integration period of the eye.

- **Light Adaption** : the human eye is sensitive to light intensity over a remarkable range. This is due to the range of photoreceptors in the retina as well as optical factors such as pupil dilation. Sekuler and Blake (1994) offer the example that when entering a dark cinema your light sensitivity can improve by a factor of around 100,000. The level of an observer's light adaption is controlled by the degree of retinal illumination via a feedback system. However, different photoreceptors can take different lengths of time in order to adapt to a new light intensity, e.g. the dark adaption period for rods is around 40 minutes, but only around 10 minutes for cones.

- **Display Factors** : if we are viewing images on a display device then the brightness, contrast, colour, and gamma settings of that display will affect the appearance of any stimuli to which we are exposed.

Individual Considerations :

- **Age** : contrast sensitivity varies as a function of age. For example, an infant's CSF is significantly displaced from an adult's CSF. Consequently we know that infants can only see large, high contrast objects (Banks, 1982).

  Owsley *et al*. (1983) investigated the contrast sensitivity of adults over a range of ages (20 to 80 years). They found that contrast sensitivity degrades notably over this range, such that an 80 year old person will be unable to perceive many of the high spatial frequencies that a 20 year old person could.

- **Colour Vision** : colour is not perceived equally by everyone. Many people suffer from colour blindness of one type or another. For example, Gregory (1990) states that nearly 10% of men have red–green colour deficiency, although this is extremely rare in women. In fact, Hurvich (1981) notes that Caucasian women are 10 times less likely to have colour deficient vision than men.

- **Stereoscopic Vision** : there is a surprisingly large percentage of the population who cannot perceive depth stereoscopically. That is, they cannot perceive three dimensions as a result of the disparity between the images from each eye. It has been estimated that as many as 1 in 10 people suffer from stereoblindness.

- **Lens Aberrations** : defective vision can be caused by an eye's inability to correctly adjust the focal length of its lens in order to project a fixated object exactly onto the retina (a process referred to as **accommodation**). For example, the lens of a person with **myopic** vision will cause light to converge at a point before the retina. This can cause distortions or blurring of an image, often requiring corrective spectacles.

- **Emotional State** : the emotional state of the observer affects the dilation of their pupils. A smaller pupil size implies that less light can reach the retina. This reduction of retinal illumination will cause a drop in the observer's visual acuity (Campbell and Green, 1965).

- **Experience** : Gregory (1990) suggests that our perception of objects may be influenced by *a priori* knowledge and past experience. For example, he speculates that common objects such as oranges and lemons develop a more rich and natural colour once they have been recognised as such.

6.3.3 Other Relevant Perceptual Material

This final section will look at a number of relevant topics within the field of visual perception and discuss how these relate our work. We do not pretend to provide conclusive solutions in all of these cases: each of the selected topics might deserve an entire thesis in their own right. However we will offer tentative solutions were possible, and highlight areas of further research where appropriate.

6.3.3.1 Eye Tracking

We know that if we want to make judgements about a user's perception of a scene, then we need to know what they are looking at. We must therefore use some form of eye tracking technology in order to monitor the user's oculomotor activity. In addition, some form of 6 DOF (degrees of freedom) tracker will normally be required in order to monitor the position and orientation of the user's head, and thus resolve the combination of head and eye movements. Such head trackers are already commonly used in immersive VR systems, so we will concentrate on the area of eye trackers as these are less pervasive in the field of computer graphics.

Modern eye tracking devices generally offer angular resolutions of less than 1 deg of arc, with possible lags in the order of milliseconds. However, many eye tracking technologies are too restrictive to be practically incorporated 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[2]:

- **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 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 working ranges (around 170 deg), they are susceptible to noise and drift, and are of questionable worth for accurate gaze tracking.

At the current time, eye tracking technology presents a number of problems as far as the VR researcher is concerned. These include being relatively expensive, susceptible to inaccuracies, or requiring frequent recalibration. However, as the technology matures, these problems will hopefully diminish. Specifically, we suggest that the following criteria should be addressed in order to facilitate the effective integration of eye tracking technology into a VR system:

[2]The interested reader is referred to Ciuffreda and Tannen (1995), and Young and Sheena (1975) for a general study of eye tracking technologies, and Stampe *et al*. (1993) for a more specific discussion of the issues relevant to computer interfaces such as VR.

1. **Interoperability** : the eye tracker must be able to co-exist with the selected 6 DOF tracker(s), and it should be able to function as part of an immersive system, e.g. inside a head-mounted display.

2. **Freedom of Movement** : many eye tracking technologies require that the observer's head be restrained, e.g. using a bite bar. Obviously in a VR system the user should be able to move their head freely and so any eye tracker must be able to operate on a moving subject.

3. **Low Lag** : VR is obviously a time-critical technology. We therefore require an eye tracking technology which can operate at real-time update rates.

4. **Persistence** : the eye tracker should be resilient to drift and not require frequent recalibrations.

It would also be useful to investigate the actual merits of integrating eye tracking technology into a VR system. For example, Watson *et al.* (1995) suggest that for their search task experiment, eye tracking would not offer any appreciable user performance gain over the situation where head tracking alone is used. As a possible explanation of this, Hitchner and McGreevy (1993) note that whenever the user evokes a large change in their point of fixation, there will normally be an associated head movement. As a result their resting gaze will generally relate quite closely to their head orientation. There is some perceptual evidence to support this claim: e.g. Bahill *et al.* (1975) report that most individuals will make a combined head and eye movement to focus on objects which are outside the central 10–15 degrees of their fixation field. It is therefore likely that, for certain applications, eye tracking would be an unnecessary complication; particularly in an immersive system where head tracking is already available.

6.3.3.2 Saccadic Suppression

A **saccade** is a rapid movement of the eye which is made in order to fixate a target onto the fovea (the name comes from the French verb *saccader*, which means 'to jerk'). Saccades can occur at velocities of up to 800 deg/s and last for periods of many milliseconds[3]. The term **saccadic suppression** is used to

[3] A good rule of thumb for the duration (ms) of a saccade is: 20 plus twice the angular distance travelled (Robinson, 1964), e.g. a 10 deg saccade will last about 40 ms.

describe the phenomenon that during a saccade we do not experience blurred vision, even though our eyes are moving at very high velocities; that is, we do not appear to perceive detail during a saccade. Sekuler and Blake (1994) collate a number of the reasons which have been postulated for this effect.

During the months of February and March 1994, there was a discussion on the USENET newsgroup `sci.virtual-worlds` into the possibility of reducing the detail of a VE during a saccade; under the assumption that the visual system might not be able to detect this change. However, it is not clear exactly to what extent we should reduce detail. It is certainly not true that our visual system 'shuts down' during a saccade—if we replace a scene with a black backdrop during a saccade then we do notice this.

Interestingly, Ohshima *et al*. (1996) recently built a system that takes advantage of saccadic suppression. In their approach, the rendering process is simply suspended whilst the angular velocity of a user's gaze movement exceeds 180 deg/s. However, Ohshima *et al*. do not offer any comment on how visually effective this was, or why they chose the value of 180 deg/s. Further work and experimentation is needed in this area.

6.3.3.3 The Blind Spot

The **blind spot** is the area of the retina where all of the axons of the retinal ganglion cells meet to form the optic nerve (see Figure 1.4, Page 15). There are no photoreceptors in this region, so we cannot detect any light which falls on the blind spot. This therefore raises the question: could we reduce the detail of objects that fall onto a user's blind spot?

The angular size of the blind spot is quite large: around 5 deg (Andrews and Campbell, 1991). However, because LOD implies a per-object modulation, we would require all of an object to be projected onto the blind spot before we could degrade its detail (or perhaps even remove it completely from the scene). For an object to be small enough to project completely within the blind spot, it would have to be very small on the display device; in which case, the size-based component of our model will very probably have already reduced the object's LOD. Also, the probability of an object being small enough on the display and also in the correct location to project onto one of your retinae's blind spots will normally be low enough to make it computationally unviable.

Finally, all of this is irrelevant under normal stereoscopic vision anyway because the blind spots for both eyes are in different regions of our visual hemisphere—i.e. we actually have two blind spots. Therefore, any part of a scene which is within one eye's blind spot will always be visible to the other eye (this is why you are always instructed to close one eye when attempting to experience the loss of vision around the blind spot). We should therefore not attempt to reduce the detail of objects when they are within the user's 'blind spot'.

6.3.3.4 Maximum Spatial and Temporal Disparities: $\mathbf{D_{max}}$ and $\mathbf{T_{max}}$

The term $\mathrm{D_{max}}$ is used to represent the largest spacing between two successively presented stimuli such that an observer will perceive the sequence as a single moving stimulus, rather than two discrete stimuli. The term $\mathrm{T_{max}}$ is the temporal equivalent of this spatial phenomenon. These values are therefore important in order to understand how we may preserve the illusion of apparent motion in a computer graphics system.

It is known that $\mathrm{D_{max}}$ is a function of stimulus size and density (Eagle and Rogers, 1991), and that this value increases with retinal eccentricity. However, little work has been done to assess the values of $\mathrm{D_{max}}$ and $\mathrm{T_{max}}$ for complex, real-world images: most of the work in this field has been restricted to simple random dot displays. In our model we therefore make no quantitative attempt to measure $\mathrm{D_{max}}$ and $\mathrm{T_{max}}$ for an object and use these values to ensure the perceptually continuous presentation of the object. However, qualitatively, the act of improving frame rate by reducing detail will increase the likelihood that these thresholds are not exceeded, and therefore that object motion is perceived less discontinuously.

6.3.3.5 Hyperacuity

The term **hyperacuity** is used to describe the paradoxical phenomenon that certain stimuli can be perceived which are smaller than the size of a single photoreceptor cell. Photoreceptors in the fovea subtend around 25–30 sec of arc (which corresponds to a maximum spatial frequency of around 60 c/deg). However, it has been shown that it is possible to discriminate the non co-linearity of two thick abutting lines to a resolution of 2–5 sec of arc (this is referred to as **vernier acuity**). We must therefore ask ourselves: are we being over simplistic by only using contrast sensitivity to measure perceived detail?

Hyperacuity is thought to be caused by differences in the mean distribution of light sampled over a number of photoreceptors (Morgan, 1991). The effect is therefore dependent upon the large spatial spread over which two adjacent features extends, i.e. any isolated feature which is smaller than a single receptor will still remain undetectable. We can therefore see that hyperacuity merely gives us a higher positional accuracy between adjacent features (discrimination), but it does not increase the fundamental resolution limit of our visual system (detection).

Our model is concerned with the *detection* of features—i.e. deciding whether an entire feature is no longer resolvable by a user—and is therefore unaffected by hyperacuity. Also, hyperacuity is largely confined to low velocity targets at the fovea. For example, Levi *et al*. (1985) and Schor and Badcock (1985) report that hyperacuity performance degrades markedly with eccentricity, more so than contrast sensitivity. In conclusion then, hyperacuity is an interesting curiosity, but not one which should affect the accuracy of our current contrast sensitivity based model.

6.3.3.6 Temporal Antialiasing

Just as spatial antialiasing attempts to smooth computer images over space, temporal antialiasing attempts to smooth images over time, i.e. to blur objects as a function of their velocity. This is done in order to simulate the effect that we perceive as fast moving objects race past our point of fixation. It would be fair to say that this a more technological rather than perceptual topic, however it does raise the question: what are the perceptual parallels between temporal antialiasing and the temporal optimisations effected by our model?

The most important distinction to make is that temporal antialiasing techniques attempt to *add* cues to an image in order to mimic characteristics of the visual system; whereas our velocity LOD model attempts to *remove* detail from a scene by taking advantage of limitations in the visual system. The similarity between the two techniques is that they both remove high spatial frequency components within an object; although temporal antialiasing does this by blurring the image of an object, and velocity LOD does this by using less complex geometry for the object.

Historically, most temporal antialiasing techniques have been developed for non real-time applications (such as ray tracing) and are therefore too computationally expensive for use in a real-time VR system (Williams, 1993); however, a number of real-time techniques are beginning to emerge. The following list presents a few of the techniques that have been developed thus far:

- **Super-Sampling** : for each frame, multiple images are generated over time and then merged together using a simple weighted sum averaging algorithm, e.g. a box filter (Korein and Badler, 1983).
- **Distributed Ray Tracing** : the various reflected and transmitted rays of a ray tracer are distributed in time as well as space, taking into account any changes in visibility and shading (Cook *et al.*, 1984).
- **Fourier Techniques** : a Fast Fourier Transform (FFT) is performed on the image and then convolved with a Point-Spread Function (PSF) which describes the image path of the projected motion. The inverse FFT is then calculated to produce the blurred image (Potmesil and Chakravarty, 1983).
- **Practically Frameless Rendering** : only a fraction (e.g. a quarter) of all pixels are updated each frame. By distributing these pixels uniformly, the entire image appears to update progressively over a number of frames; producing a 'dissolving' effect. Wloka *et al.* (1995) proffer this as a crude motion blur technique.

In general, temporal antialiasing is a desirable feature because it would allow rapidly moving objects to be perceived more faithfully (e.g. failing rain would appear as streaks rather than individual drops). However, as long as the implementation of such features could degrade the performance of a VR system, then they must be considered contrary to the goals of this thesis.

6.4 Summary

We have covered a wide range of topics in this chapter; with our perspective ranging from that of a computer scientist, to that of a vision scientist. As a result, we hope that we have addressed many of the questions which may have occured to the reader; and hopefully alerted them to further issues and considerations in the process.

We began our discussion with an examination of our model for perceptually modulated LOD. This was done in reference to the four major problems that were identified at the start of this thesis. We found that our model provides satisfactory solutions to all of these problems. Following from this we illustrated how the results from the various psychophysical studies of Chapter 5 could be used to produce a revised and more efficient model for highest visible spatial frequency (α_{vis}). We then discussed and contrasted a number of the more influential design decisions behind our model. Finally, we showed that our model could be easily extended to support fixed frame rate LOD. This generalises our model further such that it encompasses all forms of LOD modulation which are currently employed in real-time computer graphics systems.

Our next perspective was to look at the general issues facing a perceptually modulated LOD system. We noted that such a system would be more applicable to an immersive VR system; also that on the topic of level of detail, it would be inadvisable to mix various rendering models; and that when producing LOD, some mechanism should be employed to limit the spatial extent of a simplification. We also developed a couple of examples to illustrate the actual effects of a perceptually modulated LOD system in a real-world scenario. This showed that using either eccentricity or velocity optimisations on their own offers little advantage. However when used together, they represent a valuable tool for reducing detail.

In the final portion of this chapter we were concerned with looking at our model from a vision scientist's perspective. We drew the readers attention to the inherently subjective nature of perceptual theories and illustrated this in terms of our work by describing many of the factors affecting a user's visual perception. To complete our discussion we presented a cross-section of other relevant areas within the field of visual perception and illustrated how these relate to the topic of this thesis.

Upon reaching this stage, we have completed the main body of our work. We have described the background work and theories of our area, developed suitable tools to aid us in our task, produced a model and a subsequent implementation for perceptually modulated LOD, built a prototype system and evaluated its efficacy, and finally we have instigated a thorough discussion on the issues and implications of such systems. We are now in a position to draw conclusions on the merit and validity of this research programme.

Chapter 7

Conclusions

'There are more things in heaven and earth, Horatio,
than are dreamt of in your philosophy.'

(William Shakespeare, *Hamlet*)

This final chapter will consolidate all of the material presented thus far and focus the aims and achievements of this treatise. This will involve presenting conclusions on our work in the area of perceptually modulated LOD, and explicitly highlighting the original contribution to knowledge which has been made. In addition to this, we will offer a number of topics for further research. This will conclude the main body of our thesis.

7.1 Assessment of Perceptually Modulated LOD

The single driving ambition of this dissertation has been to reduce lag in VR systems by optimising the graphical content of objects without affecting the quality of images perceived by the user. This has been achieved. Specifically, let us note the following summaries:

1. We have shown how to design a perceptually modulated LOD system using sound knowledge from the field of visual perception. Furthermore, we have shown that an efficient implementation of this can be produced for use in a real-time computer graphics system.

2. We have also shown that perceptually modulated LOD can substantially reduce lag in a VR system and subsequently improve user performance, e.g. in our test application, we gained up to a five-fold improvement in frame rate, a three-fold improvement in the accuracy of users to perform the task, and an improvement in user response time of $\times\ 1\frac{2}{3}$.

With the results that were obtained from our implementation of a perceptually modulated LOD system, and the subsequent discussion that was performed, we can present the chief findings of this thesis as follows:

- We submit that a complete implementation of perceptually modulated LOD is more applicable to an immersive VR system than a desktop one. This is due to the following reasons:

  1. The display can only be optimised for a single individual because two users viewing the same scene could be looking at different regions of the screen and scanning at different speeds. Using a head-mounted display circumvents this ambiguity.
  2. Most immersive systems employ some form of head tracking. This is not normally available on the desktop and so we can never know where the user is actually looking in a desktop scenario. The best we can do in the latter case is to instantiate the focus point to a fixed point on the display.
  3. In a head-tracked system, velocity LOD will be particularly beneficial for optimising the detail of a scene during periods of high user head motion. This is not possible under desktop conditions.
  4. The visual environment within a head-mounted display is much more predictable and controllable than for a computer monitor sitting on a desk in an office. We are therefore less concerned about effects such as background illumination, glare, reflections, etc. affecting the user's percept.

- From the examples that were provided in Section 6.2.2, it is apparent that employing either eccentricity or velocity LOD on their own will prove unprofitable (this is particularly so for immersive systems using an LCD based HMD). However, if these two components are combined, then the product will be significantly greater than the sum of the individual contributions (refer to Figure 6.3). We therefore submit that eccentricity and velocity optimisations, if used at all, should be used in combination.

- It appears that, even when combined, eccentricity and velocity contribute to the total speedup to a relatively minor extent (an average of $\sim$10% for our test application: see Figure 5.16). We can therefore state that traditional size (or distance) based LOD methods provide the largest opportunity to improve the performance of a system.

  Note however that this result is only applicable under situations where the user's head is static. Under dynamic head movements the velocity component could potentially offer a far greater contribution to the reduction of detail in a scene.

- It is important to appreciate the modeling issues associated with perceptual LOD. Our goal is to change the geometrical complexity of objects based upon perceptual judgements. The LOD generation software should reflect this fact by incorporating some mechanism to specify the scale of a simplification. We therefore submit that, to be useful, all LODs should exhibit a range of visual complexities as well as a range of geometrical complexities, i.e. as the LOD of an object decreases, there should be a corresponding drop in the spatial frequency content of each model.

7.2 Thesis Contribution

Now that we have summarised the conclusions of this thesis, we will explicitly highlight how our work makes an original contribution to knowledge. Accordingly, the following seven points are offered as the principal scholarly contributions that have been made by this thesis:

1. A thorough literature review of the field of level of detail, including the process of polygon simplification.

2. The identification of a relevant metric from the visual psychology literature to describe and model the perception of spatial detail.

3. A demonstration of the applicability of this metric to the field of real-time, full-colour computer graphics.

4. The development of a computer algorithm to assess the perceptual content of a computer-generated scene in terms of this metric.

5. The collation and extension of mathematical models of human contrast sensitivity to form an efficient, machine-computable estimate of the visibility of detail; incorporating effects of angular velocity and retinal eccentricity.

6. The development of a polygon simplification framework to support the use of level of detail in a perceptually predictable manner.

7. The description, implementation, and evaluation of a perceptually modulated level of detail system based upon the above work.

Perhaps the most valuable contribution this thesis makes is with regard to the cross-disciplinary application of theory from the field of visual perception to the field of computer science. This in itself is no minor feat because the visual perception literature is voluminous and, as Ware and Knight (1995) comment, often couched in arcane jargon. It is therefore hoped that this work has helped to make the results of vision research more accessible to computer scientists, and to illustrate how knowledge from perceptual psychology may usefully be applied to our domain.

7.3 Further Work

To complete the thesis we will present a few suggestions for further research. We have covered a lot of material in these pages; spanning topics such us image segmentation, colour vision, Fourier analysis, physiology, polygon simplification, threshold vision, system latency, psychophysics, and level of detail. There is therefore much that could be investigated furthered. However, to contain the discussion, we have selected the following four topics which we feel offer the most potential for expansion.

7.3.1 Improved Metrics for Assessing Perceived Detail

If there is only one problem which has delayed the development of a perceptually modulated LOD system before now, then it is the fact that no complete theory currently exists to explain how we perceive and differentiate arbitrary stimuli. In this thesis we have proposed and implemented a simplistic scheme using colour difference theory to segment an image. This appears to give reasonable results, but we suggest that more work could be done to extend or replace this particular technique, and hence to produce a more perceptually robust solution. For example, even if the notion of using colour differences is assumed valid, much work needs to be done to improve the accuracy and generality of current colour difference theory.

Given the nature of such an activity, this research would be best furthered by a body with cross-disciplinary skills in visual perception and computer graphics. Of course, the development of a general model of perceptual image segmentation might be too complicated at the current time; however, if we restrict ourselves to the genus of computer-generated images, then a suitable solution may present itself.

It is important to note that the development of such a metric would have value transcending this single application. For example, we could use this technique to compare the perceptual impact of two different rendering algorithms, or analyse the effect of introducing various antialiasing techniques. Alternatively, such a tool could be employed by perceptual psychologists to further analyse the intricacies of our visual system using the power and flexibility of current computer graphics technology.

7.3.2 Transparent LOD Generation

One of the achievements of this work has been to remove all subjectivity from the application of LOD techniques by basing these on sound principles of visual perception. The designer of a VE would therefore not need to worry about which values to supply for the switching thresholds between successive models: this is done automatically. However, on the modeling side, there is still an element of subjectivity because the designer has to create some number of LODs by supplying certain parameters to a polygon simplification algorithm. The ideal situation would be if the designer had a complex model that they wanted to use in a VE, and then the system would automatically produce a good suite of LOD models, with no direction required on the designer's behalf.

Associated with this is the need for polygon simplification techniques which complement a VR system. We noted in Section 2.2.4.2 (Page 49) that many current techniques do not address all of the issues inherent in a VR system. We require techniques which are developed to be useful in our domain, rather than blindly using methods from other fields. Furthermore, these techniques should incorporate a mechanism to specify the scale of a simplification in order to constrain the perceptual effect of reducing an object's detail.

One recent development in this area is worth particular comment. Luebke (1996) presents a system in which the entire scene is stored in an octree data structure; so that volumes further down the structure represent smaller volumes of the world space (and hence finer detail). When the system decides that a particular volume level in the octree is below threshold (e.g. projects to less than a pixel), then all of the vertices in that volume are collapsed to a single point, discarding any degenerate polygons that arise. The reader will recognise the conceptual parallels to the work of Rossignac and Borrel (1992). This global, real-time view to optimising detail offers many benefits. Firstly, it requires no intervention on the part of the VE designer. Secondly, it has the ability to merge groups of adjacent objects into a single entity which could not be done under an LOD paradigm.

7.3.3 An Eye Tracking Feasibility Study

On a number of occasions we have acknowledged that eye tracking is an essential component of any true perceptually modulated LOD system. However the remit of our work has not enabled us sufficiently to investigate the practicalities of this conjunction.

Our prototype evaluation was performed in such a manner that the observers' gaze was restricted to a known spatial location. Our results should therefore be applicable to any system where the observer's gaze position is known. However, there are a number of issues associated with the integration of an eye tracking facility into a perceptually modulated LOD system which should be explicitly investigated. We therefore suggest that the following would be a purposive direction of research activities.

- We require more information about the availability and performance of plausible eye tracking technologies for use in immersive VR systems. These devices have invariably been used for clinical applications in the past: we require an appreciation of the field from our perspective. Also, we need to know how active the field is, and how the technology might mature over a 5–10 year time frame.

- It would be illuminating to investigate our perceptually modulated LOD techniques in a system which supports eye tracking. For example, how do factors such as the tracker's dynamic response, resolution, accuracy, and drift affect the systems ability to modulate detail with no perceptual defect? And how is this likely to change as eye tracking technology matures?

- It is important to know the extent to which eye tracking is actually required. We have already suggested that for some applications, head tracking might be sufficient. Work needs to be done to investigate the various cost, performance, and benefit implications of utilising eye tracking technologies.

7.3.4 A General Sensory Paradigm

Throughout this thesis we have focussed purely on the issues concerning our visual sensorium. However, it is interesting to note certain parallels with other senses. For example, in the field of auditory perception there are curves known as **audibility functions** (AFs) which describe the range of tone frequencies that a normal young adult can hear. Similar descriptions exist in the field of tactile perception to describe the touch sensitivity and acuity of individuals. In fact, in the latter case, devices known as tactile gratings are used which contain a series of grooves of a particular depth and spatial frequency. The analogy to contrast gratings and contrast sensitivity functions should be obvious.

We therefore raise the question of whether the perceptual optimisations we have developed here could be extrapolated to apply to other senses. This is a purely speculative suggestion: the author has not investigated the feasibility or value of such an effort. However, it is offered here as an interesting extension to our work, and one which may warrant further investigation.

Appendix A

Just Noticeable Difference Evaluations

> *'If I have seen farther than others,*
> *it is because I was standing on the shoulders of giants.'*
>
> (Sir Isaac Newton)

This appendix presents the results from a series of psychophysical experiments which were performed to investigate the efficacy of the CIELUV colour difference equation (CIE, 1979). The experimental procedure is described by Pokorny *et al*. (1991) and proceeded as follows:

> A bipartite display was generated on a 20" Hitachi CM2198MSG computer monitor using an SGI RealityEngine^2 workstation. The two fields of the display occupied 2 degrees of the observer's visual field and were presented on a neutral-hued (mid-gray) background in order to fix the observer's state of adaption. The experiment was performed in an environment with high (photopic) light conditions and the observer was photopically adapted with corrected-to-normal vision.
>
> For each test, the subject was presented with an initial colour in both fields of the bipartite display. The subject was then free to increment the hue or saturation of the right-hand field until a perceivable difference was observed. At this point, the CIELUV colour difference (ΔE^*_{uv}) was noted for the two colours.

The experiment was repeated for a number of initial colours, and a number of increment hues and saturations, in order to evaluate the colour difference equation over a wide range of chromatic stimuli.

The following table presents a summary of the results obtained. The first two columns describe the initial colour of the bipartite fields and the increment colour which the observer was free to alter. The third columns presents the RGB specification of the initial colour; while the fourth column records the final RGB value of the right-hand field once the observer had located a noticeable difference. The last column presents the CIELUV colour difference value between the latter two RGB values.

Initial Colour	Increment Hue	Initial RGB	Final RGB	ΔE^*_{uv}
Red	Red	128,0,0	136,0,0	**3.93**
		128,0,0	120,0,0	**4.10**
		255,0,0	244,0,0	**3.54**
Green	Green	0,128,0	0,134,0	**2.33**
		0,128,0	0,121,0	**2.82**
		0,255,0	0,244,0	**2.79**
Blue	Blue	0,0,128	0,0,140	**4.85**
		0,0,128	0,0,117	**4.72**
		0,0,255	0,0,242	**3.48**
Cyan	Cyan	0,128,128	0,134,134	**1.67**
		0,128,128	0,123,123	**1.43**
		0,255,255	0,245,245	**1.81**
Magenta	Magenta	128,0,128	136,0,136	**3.10**
		128,0,128	120,0,120	**3.21**
		255,0,255	247,0,247	**2.00**
Yellow	Yellow	128,128,0	134,134,0	**2.10**
		128,128,0	122,122,0	**2.13**
		255,255,0	245,245,0	**2.23**
Black	Red	0,0,0	11,0,0	**29.02**
Black	Green	0,0,0	0,7,0	**28.20**
Black	Blue	0,0,0	0,0,15	**16.00**
Black	Cyan	0,0,0	0,8,8	**22.66**
Black	Magenta	0,0,0	11,0,11	**26.74**
Black	Yellow	0,0,0	7,7,0	**27.00**
Black	White	0,0,0	7,7,7	**19.00**

Continued on next page ...

Initial Colour	Increment Hue	Initial RGB	Final RGB	ΔE^*_{uv}
White	Red	255,255,255	235,255,255	**5.10**
White	Green	255,255,255	255,246,255	**3.43**
White	Blue	255,255,255	255,255,242	**4.46**
White	Cyan	255,255,255	255,247,247	**2.30**
White	Magenta	255,255,255	243,255,243	**4.42**
White	Yellow	255,255,255	247,247,255	**3.00**
Red	White	255,0,0	255,13,13	**28.82**
		128,0,0	134,6,6	**16.13**
Green	White	0,255,0	33,255,33	**25.00**
		0,128,0	5,133,5	**4.43**
Blue	White	0,0,255	15,15,255	**10.14**
		0,0,128	8,8,128	**11.54**
Cyan	White	0,255,255	41,255,255	**12.89**
		0,128,128	8,136,136	**3.18**
Magenta	White	255,0,255	255,20,255	**15.72**
		128,0,128	124,6,134	**5.19**
Yellow	White	255,255,0	255,255,45	**22.50**
		128,128,0	133,133,5	**2.84**
Red	Blue	255,0,0	255,0,24	**28.86**
		128,0,0	128,0,17	**28.73**
Red	Green	255,0,0	255,10,0	**17.38**
		128,0,0	128,7,0	**16.79**
Green	Blue	0,255,0	0,255,21	**13.04**
		0,128,0	0,128,17	**15.65**
Green	Yellow	0,255,0	25,255,0	**10.50**
		0,128,0	6,134,0	**3.36**
Blue	Yellow	0,0,255	15,15,255	**10.14**
		0,0,128	8,8,128	**9.37**
Blue	Magenta	0,0,255	16,0,255	**6.94**
		0,0,128	8,0,136	**7.50**

Table A.1: *CIELUV colour difference values for a range of chromatic stimuli.*

Appendix B

Example Spatial Frequency Segmentations

'If I have not seen as far as others,
it is because giants were standing on my shoulders.'

(Hal Abelson)

This appendix contains sample results produced by our image segmentation technique for extracting the perceived spatial frequencies within a digital image (see Section 3.1, page 67). This consists of a collection of images, each accompanied by an illustration showing the extent of all visual features found by the segmentation procedure. (Note that all computer-generated images were produced in full-colour, despite being rendered in greyscale for presentation in this thesis.)

We will begin by presenting the four levels of detail which were used for the user task performance study (see Section 5.2, page 147). For each of these we will illustrate the result of the feature extraction process and the subsequent spatial frequency calculation. Following from this, we will present a number of further example feature extractions, and also an example of the feature extraction process for an object which is rendered in flat-shaded, smooth-shaded, and texture mapped modes.

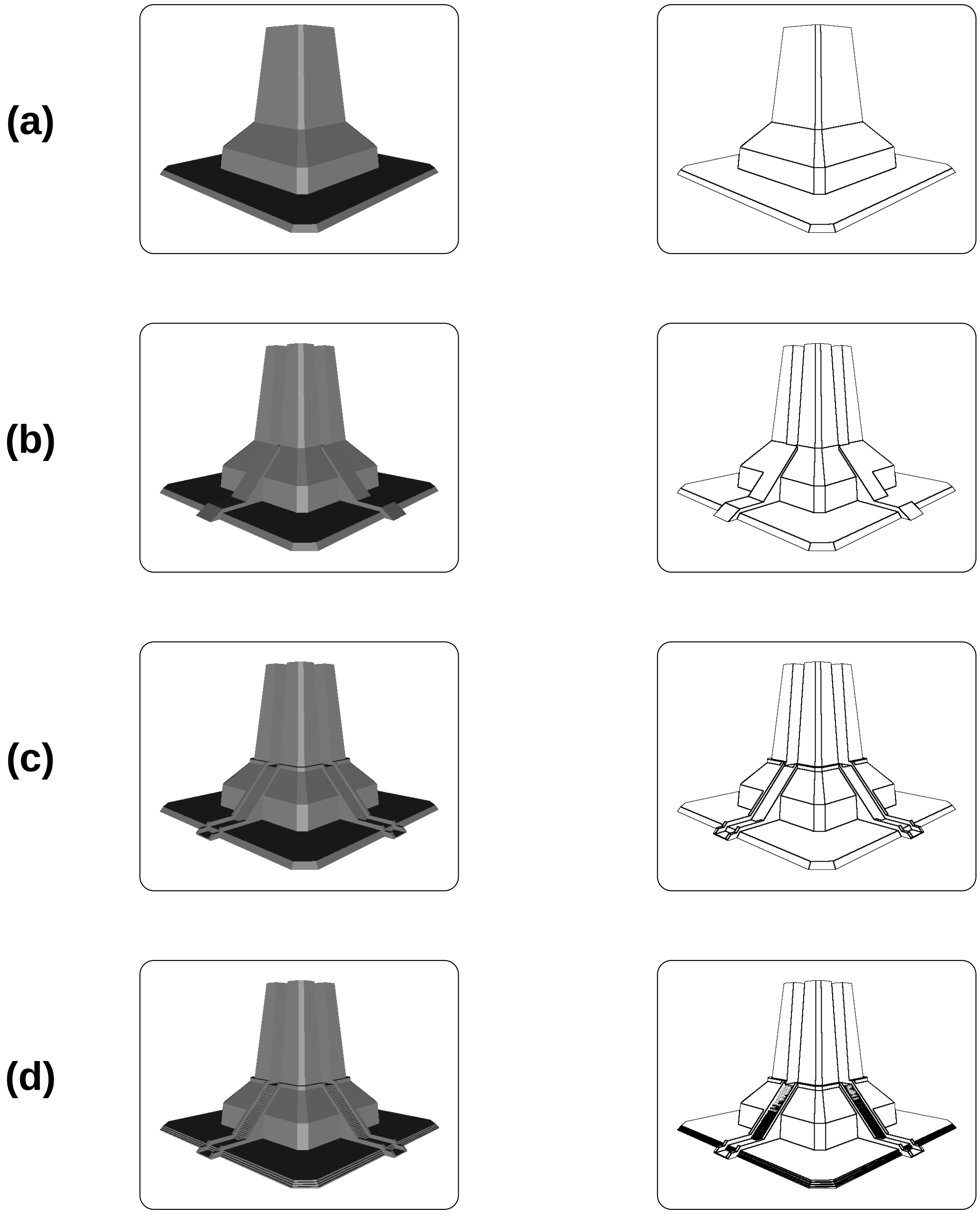


Figure B.1: *Feature segmentations for each of the 'temple' LODs which were used for the user task performance study. The figures on the left present the rendered image of each model, while the figures on the right illustrate the extent of all features which were located in the adjacent image.*

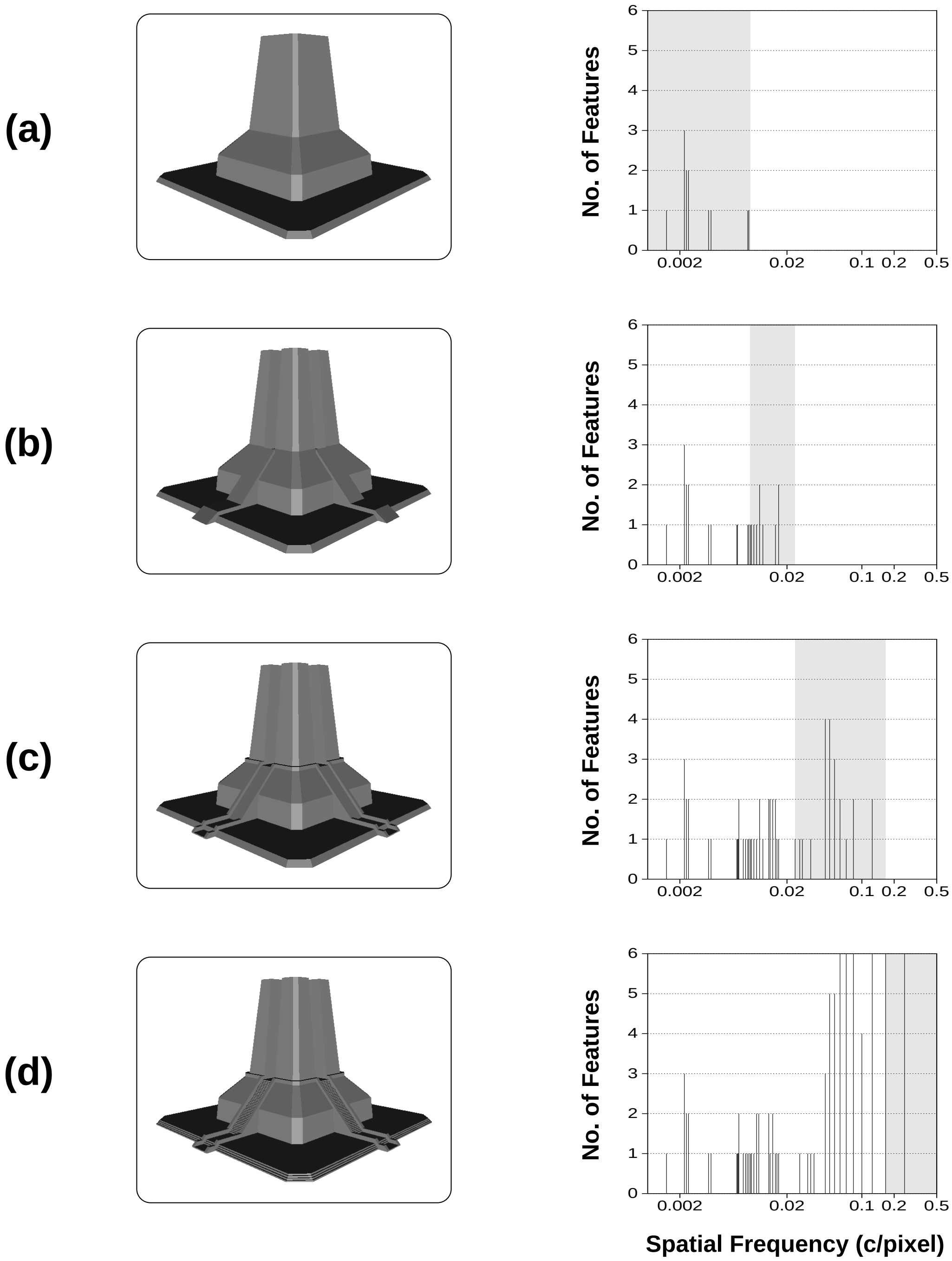


Figure B.2: *Relative spatial frequency (c/pixel) profiles for each of the 'temple' LODs. The shaded regions on each graph represent the frequencies over which that model can be used without defect (see Page 116). Note that in (d), the data have been thresholded to a maximum of 6 features in order to maintain readability and scale consistency between each graph.*

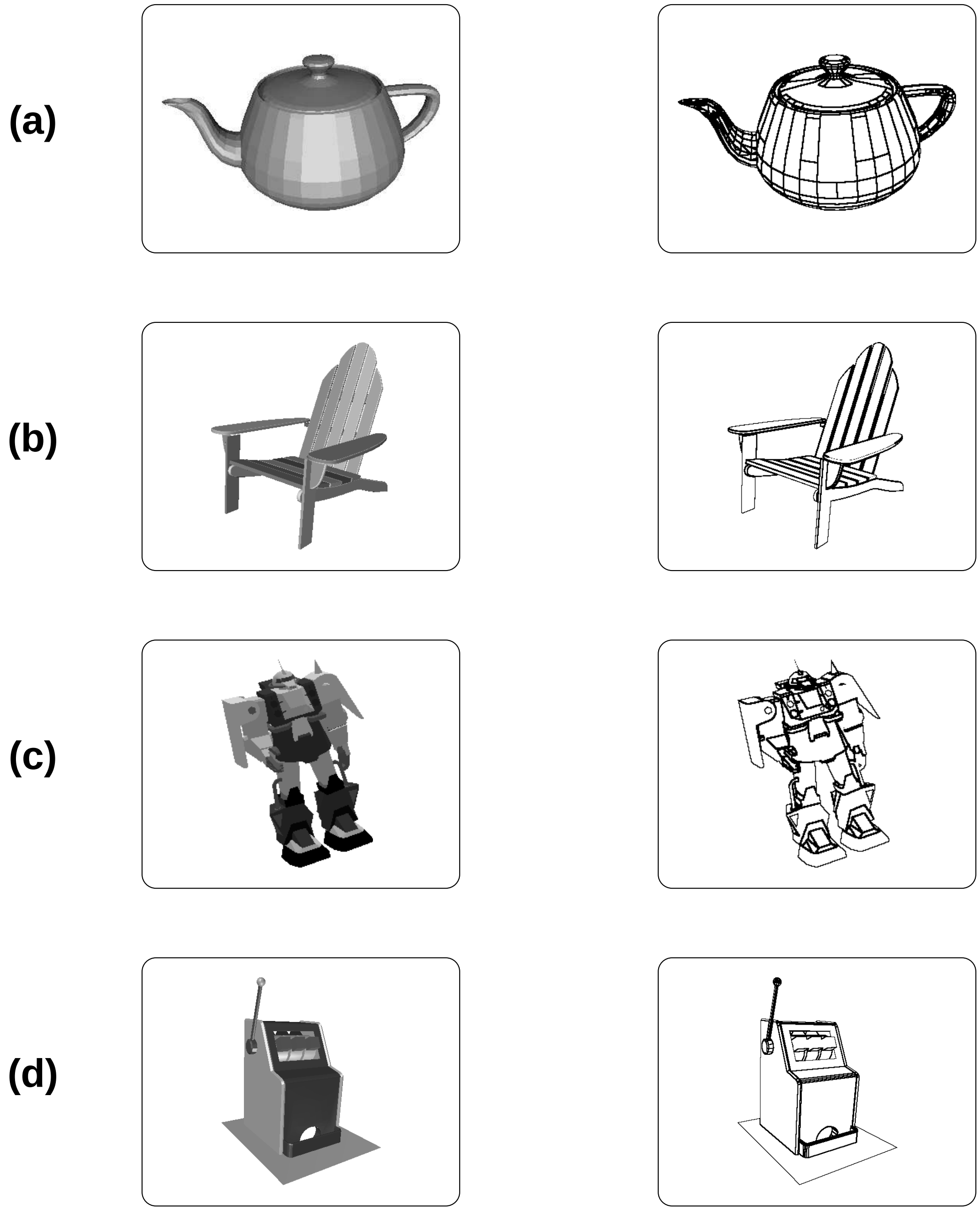


Figure B.3: *Feature segmentations for various objects: (a) the ubiquitous 'teapot', (b) 'chair', (c) 'robot', and (d) 'slotMachine'. Models (a) and (c) were flat-shaded, while (b) and (d) were smooth-shaded.*

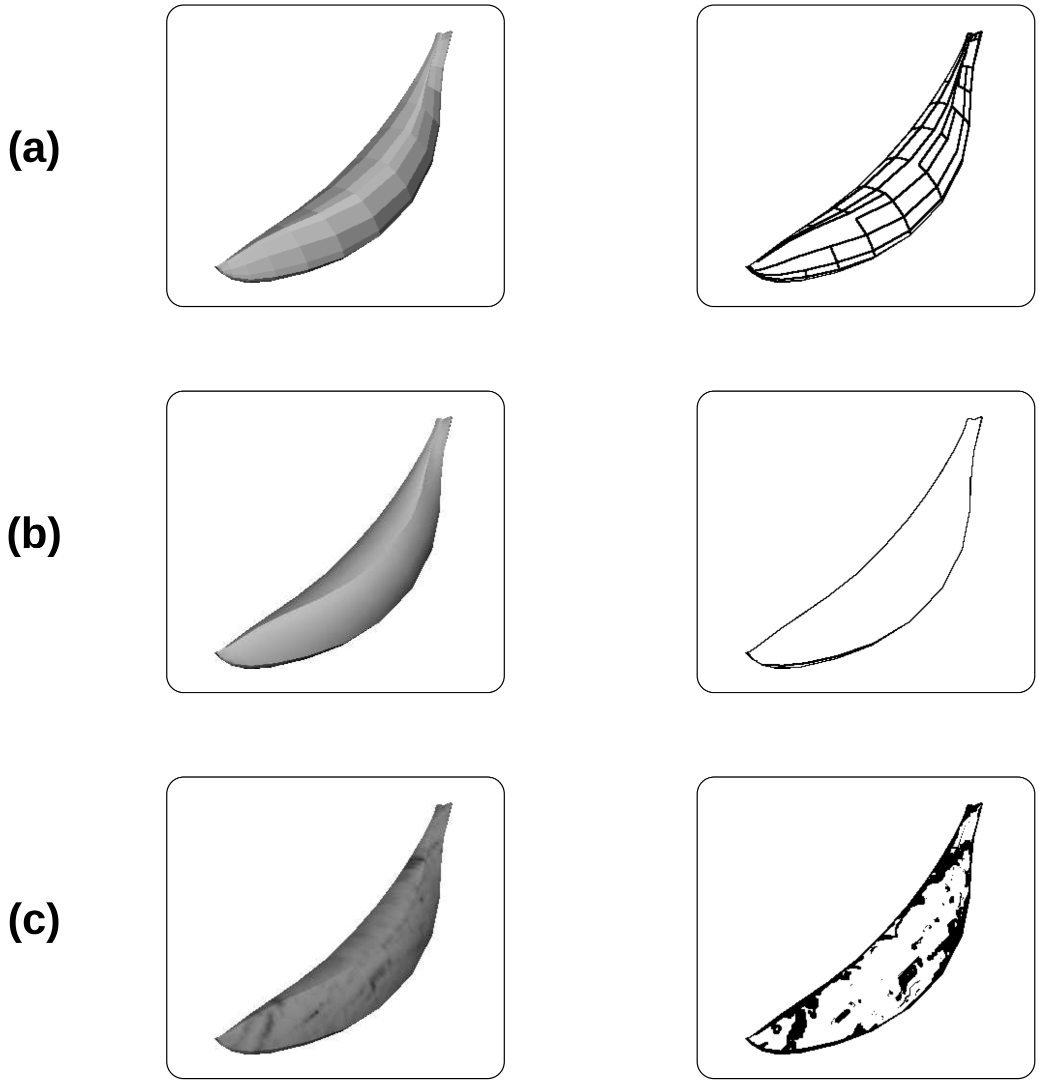


Figure B.4: *Feature segmentations for a single object under different rendering modes: (a) flat-shaded, (b) smooth-shaded, and (c) texture mapped. Note the dichotomy that the segmentation of (b) is far less complex than (a), but smooth-shading is generally more computationally expensive than flat-shading (see Section 6.2.1.2, page 170).*

Glossary of Terms

'Do not go gentle into that good night,
Old age should burn and rave at close of day;
Rage, rage against the dying of the light.'

(Dylan Thomas, *Do Not Go Gentle*)

2AFC : see *2 Alternative Forced Choice*.

2 Alternative Forced Choice : an experimental procedure in which the user is presented with a stimulus in one of two intervals and must identify the correct one, guessing if necessary. This overcomes the problems of subject bias found in simple Yes/No type experiments (p. 138).

blind spot : the area of the retina where all of the axons of the retinal ganglion cells meet to form the optic nerve (p. 182).

cones : a class of *photoreceptor cells* responsible for the detection of chromatic light; specialised for daylight conditions (p. 14).

contrast : the difference in light intensity between an object and its immediate surroundings.

contrast grating : a pattern of alternating light and dark bars generated by varying *contrast* sinusoidally across the display. Used to measure a subject's *contrast sensitivity* (p. 53).

contrast sensitivity : a measure of an observer's sensitivity to spatial detail in terms of its *contrast*. Defined as the reciprocal of *threshold contrast* (p. 55).

contrast sensitivity function : a curve that records an observer's ability to resolve detail in terms of *contrast* and *spatial frequency* (p. 54).

cortical magnification factor : an equation describing the disproportionate weighting of the *visual cortex* to represent the foveal region of the retina (p. 91).

CSF : see *contrast sensitivity function*.

eccentricity : see *retinal eccentricity*.

field of view : the solid angular subtense which is viewable by the eye.

focus point : the 2D coordinate on the viewport which is used to calculate the *eccentricity* of an object. For example, the point where the user is gazing (p. 120).

forced-choice method : see *2 Alternative Forced Choice*.

Fourier analysis : a method for decomposing any waveform into the series of harmonic frequencies that sum to produce that waveform.

FOV : see *field of view*.

fovea : the point of the retina which is most sensitive to detail (p. 21).

fundamental frequency : the lowest frequency component that exists in a waveform (p. 58).

geons : a theory of object recognition proferred by Biederman (1987) suggesting that recognition is based upon identifying a small number of primitive shapes in an object (p. 45).

Gestalt psychology : a school of visual perception, prevalent earlier this century, believing that overall structure is more important than individual components for form perception (p. 51).

Gestalt principles of organisation : certain stimulus properties that are believed to control the perceptual grouping of objects, e.g. proximity, similarity, closure, and good continuation (p. 51).

global topology : the geometric structure of an entire polygonal model.

grating : see *contrast grating*.

head-mounted display : a display device which is mounted on the user's head and normally incorporates some form of tracking technology so that the user's head orientation and/or position can be ascertained.

highest displayable spatial frequency : the term that we use in this thesis to denote the highest *spatial frequency* that can be produced by a display device; referred to as the symbol ξ (p. 124).

highest visible spatial frequency : the term that we use in this thesis to denote the highest *spatial frequency* that we expect the user to be able to perceive; referred to as the symbol α_{vis} (p. 123).

HMD : see *head-mounted display*.

hyperacuity : term used to descibe the phenomenon that certain stimuli can be perceived which are smaller than the size of a single retinal *photoreceptor cell* (p. 183).

hysteresis : a lag in the transition between different *levels of detail*. Often performed to stop models continuously switching at threshold (p. 29).

image segmentation : a digital image processing technique which attempts to segment an image into a number of individual regions for independent analysis.

inferior retina : the lower half-meridian of the retina.

JND : see *just noticeable difference*.

just noticeable difference : the smallest unit perceptual difference between two colours. The degree to which any two colours are perceived distinct can be expressed as a number of JNDs.

level of detail : a computer graphics technique for optimising performance whereby a number of different models are provided for a single object—each varying in complexity—and the system can select the most appropriate model to use at any juncture.

local topology : the geometric structure surrounding a particular vertex or face of a polygonal model.

LOD : see *level of detail*.

Marching Cubes : an algorithm that creates a polygon model from a *voxel* (volumetric) dataset. Developed by Lorensen and Cline (1987) (p. 42).

modulation transfer function : a measure of a display device's ability to maintain the *contrast* of a signal as a function of its *spatial frequency* (p. 126).

MTF : see *modulation transfer function*.

multichannel model : contemporary model of early vision processing stating that information in a scene is analysed simultaneously at a number of different scales (p. 52).

nasal retina : the half-meridian of the retina located nearest to the nose.

perceptually uniform colour space : a mathematical representation for colour in which the numerical difference between two colours is directly related to their perceptual distance.

photopic : referring to daytime lighting conditions.

photoreceptor cell : the cells in the retina that are responsible for converting incident light energy into neural signals. There are two classes: *rods* and *cones* (p. 14).

polygon simplification : a process that takes an original polygon model and produces another such description, retaining the general shape and appearance of the original, but containing fewer polygons (p. 40).

popping effect : the noticeable flicker that can occur when the graphics system switches between different levels of detail (pp. 12, 50).

predictive scheduler : a system which adjusts the detail of objects, in order to achieve a desired frame rate, based upon the complexity of the scene about to be rendered (p. 33).

psychophysics : the branch of psychology which is concerned with establishing quantitative relations between physical stimulation and perceptual events.

reactive scheduler : a system which adjusts the detail of objects, in order to achieve a desired frame rate, based upon whether the previous frame was rendered before or after the deadline (p. 33).

receptive field : the region immediately surrounding a cell that stimulates that cell if excited (p. 17).

retinal eccentricity : angular deviation from the centre of the retina, often taken as the *fovea* (p. 21).

retinal ganglion cells : the cells in the retina that extract gradients of light from the *photoreceptor cells*. Their outputs form the optic nerve (p. 14).

rods : a class of *photoreceptor cells* responsible for the detection of achromatic light; specialised for dim light conditions (p. 14).

saccade : a rapid movement of the eye which is made in order to fixate a target onto the *fovea* (p. 181).

smallest bounding sphere : the smallest sphere that completely encloses an object and whose centre is coincident with the centre of that object (p. 112).

Snellen fraction : a measure of *visual acuity* that uses a fraction in the form $20/n$ to relate an observer's acuity to that of a normal adult (p. 63).

spatial frequency : a measure of the rate of change of intensity across the retina; often measured in units of cycles per degree, c/deg (p. 53).

spatiotemporal threshold surface : the surface which describes the sensitivity of an observer to stimuli of varying spatial and temporal characteristics (pp. 89, 165).

spherical tessellation : the process of approximating a sphere with a mesh of polygons, usually triangles (p. 113).

staircase method : an adaptive psychophysical procedure which attempts to converge on a signal threshold by selecting new stimuli based upon the observer's previous responses (p. 139).

standard observer : a notional 'average' human for whom we can develop general models of perception that are applicable to most of the population (p. 64).

superior retina : the upper half-meridian of the retina.

temporal retina : the half-meridian of the retina which is located nearest to the temple on the same side of the face.

threshold contrast : the minimum *contrast* required to see a target (p. 55).

topology : see *local topology* and *global topology*.

transition spatial frequency : the term that we use in this thesis to denote the *spatial frequency* threshold when two successive *levels of detail* for an object can be switched between; referred to as the symbol α_{trans} (p. 116).

triangulation : the act of decomposing an arbitrary polygon into triangles.

troland : (td) a measure of light intensity defined as the product of source luminance (cd/m^2) and pupillary diameter squared (mm^2).

unique spatial frequency : the term that we use in this thesis to denote a *spatial frequency* (magnitude and position) that exists in the image of only one of two successive *levels of detail* of an object, i.e. a spatial frequency that differs between successive LODs (p. 108).

VE : see *virtual environment*.

vestibular system : the sensory mechanism in the inner ear that detects head movements and aids balance (p. 5).

view sphere : the bounding sphere around an object whose surface represents all possible viewpoints to that object for a certain distance from the centre of the object (p. 113).

virtual environment : a 3D computer-generated world that is simulated by a *virtual reality* system.

virtual reality : a technology that allows the user to particpate in, and interact with, a 3D computer-generated world.

visual acuity : a measure of the smallest detail which an observer can resolve under ideal illumination conditions (p. 56).

visual cortex : the region of the brain responsible for visual perception (p. 13).

voxel : volume element—the 3D equivalent of a pixel for volumetric datasets. Normally arranged in a fixed regular grid. (p. 43).

VR : see *virtual reality*.

window of visibility : the range of *spatial frequencies* that an observer can see, given sufficient contrast (p. 54).

Bibliography

Airey, J. M., Rohlf, J. H. and Frederick P. Brooks, J. (1990). Towards Image Realism with Interactive Update Rates in Complex Virtual Building Environments, *ACM SIGGRAPH Special Issue on 1990 Symposium on Interactive 3D Graphics*, **24**(2): 41–50.

Aliaga, D. G. (1996). Dynamic Simplification Using Textures, *UNC Technical Report No. TR96-007*, Deptartment of Computer Science, University of North Carolina, Chapel Hill, NC.

Andrews, P. R. and Campbell, F. W. (1991). Images at the Blind Spot, *Nature*, **353**(6342): 308.

Anstis, S. M. and Cavanagh, P. (1983). A Minimum Motion Technique for Judging Equiluminance in Colour Vision, *in* J. D. Mollon and L. T. Sharpe (eds), *Colour Vision: Physiology and Psychophysics*, Academic Press, London, pp. 156–166.

Astheimer, P. and Pöche, M.-L. (1994). Level-Of-Detail Generation and its Application in Virtual Reality, *Proceedings of the VRST '94 conference*, Singapore, pp. 299–309.

Atkin, P. J. (1993). Parallel Processing for Virtual Reality, *Parallel Processing for Graphics and Scientific Visualization*, University of Edinburgh.

Bahill, A. T., Adler, D. and Stark, L. (1975). Most Naturally Occuring Saccades Have Magnitudes of 15 Degrees or Less, *Investigative Ophthalmology and Visual Science*, **14**: 468.

Banks, M. S. (1982). The Development of Spatial and Temporal Contrast Sensitivity, *Current Eye Research*, **2**: 191–198.

Barfield, W. and Hendrix, C. (1995). The Effect of Update Rate on the Sense of Presence within Virtual Environments, *Virtual Reality: Research, Development, Applications*, **1**(1): 3–16.

Barten, P. G. J. (1990). Evaluation of Subjective Image Quality with the Square-Root Integral Method, *Journal of the Optical Society of America A (Optics and Image Science)*, **7**(10): 2024–2031.

Beck, J., Sutter, A. and Ivry, R. (1987). Spatial Frequency Channels and Perceptual Grouping in Texture Segregation, *Computer Vision, Graphics, and Image Processing (CVGIP)*, **37**: 299–325.

Biederman, I. (1987). Recognition by Components: A Theory of Human Image Understanding, *Psychological Review*, **94**(2): 115–147.

Billyard, A. (1993). Shifting the Software/Silicon Balance in High Performance 3D Graphics, *VR93 Virtual Reality International 93: Proceedings of the Third Annual Conference on Virtual Reality*, Meckler, London, UK, pp. 19–23.

Bishop, G., Fuchs, H., McMillan, L. and Scher Zagier, E. J. (1994). Frameless Rendering: Double Buffering Considered Harmful, *in* A. Glassner (ed.), *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, pp. 175–176. ISBN 0-89791-667-0.

Blakemore, C. and Campbell, F. W. (1969). On the Existence of Neurones in the Human Visual System Selectively Sensitive to the Orientation and Size of Retinal Images, *Journal of Physiology*, **203**: 237–260.

Borgefors, G. (1992). A Hierarchical Square Tessellation of the Sphere, *Pattern Recognition Letters*, **13**(3): 183–188.

Bracewell, R. (1965). *The Fourier Transform and Its Applications*, McGraw-Hill, inc., New York, NY.

Brigham, E. O. (1974). *The Fast Fourier Transform*, Prentice-Hall, inc., Englewood Cliffs, NJ. ISBN 0-13-307496-X.

Bryson, S. (1993). Effects of Lag and Frame Rate on Various Tracking Tasks, *Proceedings of the SPIE - The International Society for Optical Engineering*, Vol. 1915 of *Stereoscopic Displays and Applications IV*, Bellingham, WA, pp. 155–166.

Bryson, S. and Fisher, S. S. (1990). Defining, Modeling, and Measuring System Lag in Virtual Environments, *Proceedings of the SPIE - The International Society for Optical Engineering*, Vol. 1256, Bellingham, WA, pp. 98–109.

Burr, D. C. and Ross, J. (1982). Contrast Sensitivity at High Velocities, *Vision Research*, **22**: 479–484.

Caelli, T. M. and Moraglia, G. (1985). On the Detection of Gabor Signals and Discriminations of Gabor Textures, *Vision Research*, **25**: 671–684.

Campbell, F. W., Carpenter, R. H. S. and Levinson, J. Z. (1969). Visibility of Aperiodic Patterns Compared with that of Sinusoidal Gratings, *Journal of Physiology*, **204**: 283–298.

Campbell, F. W. and Green, D. G. (1965). Optical and Retinal Factors Affecting Visual Resolution, *Journal of Physiology*, **181**: 576–593.

Campbell, F. W. and Gubisch, R. W. (1966). Optical Quality of the Human Eye, *Journal of Physiology*, **186**: 558–578.

Campbell, F. W., Hulikowski, J. J. and Levinson, J. (1966). The Effect of Orientation on the Visual Resolution of Gratings, *Journal of Physiology*, **187**: 427–436.

Campbell, F. W. and Robson, J. G. (1968). Application of Fourier Analysis to the Visibility of Gratings, *Journal of Physiology*, **197**: 551–566.

Carpenter, R. H. S. (1992). Turning Vision Into Actions, *Current Biology*, **2**: 288–290.

Carter, R. (1996). Personal communication.

Carter, R. C. (1989). Calculate (Don't Guess) the Effect of Symbol Size on Usefulness of Color, *Proceedings of the Human Factors Society 33rd Annual Meeting*, pp. 1368–1372.

Carter, R. C. and Carter, E. C. (1983). CIE $L^*u^*v^*$ Color-Difference Equations for Self-Luminous Displays, *Color Research and Application*, **8**: 252–253.

Cavanagh, P. (1991). Vision at Equiluminance, *in* J. R. Cronly-Dillon (ed.), *Vision and Visual Dysfunction: Limits of Vision*, Vol. 5, MacMillan Press, Ltd., chapter 20, pp. 234–250. ISBN 0-333-52713-5.

Chrislip, C. A. and Ehlert Jr., J. F. (1995). *Level of Detail Models for Dismounted Infantry in NPSNET–IV.8.1*, Master's thesis, Naval Postgraduate School, Monterey, CA.

CIE (1979). Recommendations on Uniform Color Spaces: Color-Difference Equations, Supplement No. 2 to CIE Publication No. 15, Vienna: Commision International de l'Éclairage, Central Bureau of the CIE.

Ciuffreda, K. J. and Tannen, B. (1995). *Eye Movement Basics for the Clinician*, Mosby-Year Book, Inc., St. Louis, Missouri. ISBN 0-8016-6843-3.

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

CMC (1989). CMC: Calculation of Small Color Differences for Acceptability, *Textile Chemist and Colorist*, **21**(11): 18–21.

Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P., Brooks, F. and Wright, W. (1996). Simplification Envelopes, *UNC Technical Report No. TR96-017*, Deptartment of Computer Science, University of North Carolina, Chapel Hill, NC.

Coltman, J. W. and Anderson, A. E. (1960). Noise Limitations to Resolving Power in Electronic Imaging, *Proceedings of the Institute of Radio Engineers*, Vol. 48, pp. 858–865.

Cook, R. L., Porter, T. and Carpenter, L. (1984). Distributed Ray Tracing, *Computer Graphics*, **18**(3): 137–144.

Corless, R. M., Gonnet, G. H., Hare, D. E. G. and Jeffrey, D. J. (1993). On Lambert's W Function, *Technical Report CS-93-03*, Department of Computer Science, University of Waterloo, Canada.

Costella, J. P. (1993). Motion Extrapolation at the Pixel Level. School of Physics, The University of Melbourne.

Cowan, W. B. and Ware, C. (1985). Elementary colour coding, *SIGGRAPH 1985 Course #3 Notes: Colour Perception*, pp. 55–95.

Cutting, J. E. (1986). *Perception with an Eye for Motion*, MIT Press, Cambridge, MA.

Cutting, J. E., Springer, K., Braren, P. A. and Johnson, S. H. (1992). Wayfinding on Foot From Information in Retinal, Not Optical, Flow, *Journal of Experimental Psychology: General*, **121**(1): 41–72.

Daly, S. (1993). The Visible Difference Predictor: An Algorithm for the Assessment of Image Fidelity, *in* A. B. Watson (ed.), *Digital Images and Human Vision*, MIT Press, pp. 179–206.

Daniel, P. M. and Whitteridge, W. (1961). The Representation of the Visual Field on the Cerebral Cortex in Monkeys, *Journal of Physiology*, **159**: 203–221.

Daugman, J. G. (1984). Spatial Visual Channels in the Fourier Plane, *Vision Research*, **24**(9): 91–910.

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

Drasdo, N. (1977). The Neural Representation of Visual Space, *Nature*, **266**: 554–556.

Eagle, R. A. and Rogers, B. J. (1991). Maximum Displacement (D_{max}) as a Function of Density, Patch Size, and Spatial-Filtering in Random-Dot Kinematograms, *Investigative Ophthalmology and Visual Science*, **32**(4): 893.

Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, W. (1995). Multiresolution Analysis of Arbitrary Meshes, *Technical Report 95-01-02*, University of Washington.

Economy, R., Ferguson, R. L., Kelly, W. A. and Ramos, P. P. (1990). Continuous Terrain Level of Detail for Visual Simulation, *Proceedings of the IMAGE V Conference*, Phoenix, Arizona, pp. 144–151.

Enroth-Cugell, C. and Robson, J. G. (1966). The Contrast Sensitivity of Retinal Ganglion Cells of the Cat, *Journal of Physiology*, **187**: 517–552.

Erikson, C. (1996). Polygonal Simplification: An Overview, *UNC Technical Report No. TR96-016*, Deptartment of Computer Science, University of North Carolina, Chapel Hill, NC.

Evans, R. (1990). Image Quality Metrics and Performance of Visual Display Systems, *Proceedings of the IMAGE V Conference*, Phoenix, Arizona, pp. 115–122.

Falby, J. S., Zyda, M. J., Pratt, D. R. and Mackey, R. L. (1993). NPSNET: Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation, *Computer and Graphics*, **17**(1): 65–69. Pergamon Press, UK.

Foley, J. D., van Dam, A., Feiner, S. K. and Hughes, J. F. (1990). *Computer Graphics: Principles and Practice*, second edn, Addison-Wesley, Reading, MA. ISBN 0-201-12110-7.

Frank, L. H., Casali, J. G. and Wierwille, W. W. (1988). Effects of Visual Display and Motion System Delays on Operator Performance and Uneasiness in a Driving Simulator, *Human Factors*, **30**(2): 201–217.

Funkhouser, T. A. and Séquin, C. H. (1993). Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments, *Computer Graphics (SIGGRAPH '93 Proceedings)*, Vol. 27, pp. 247–254.

Gervais, M. J., L. O. Harvey, J. and Roberts, J. O. (1984). Identification Confusions Among Letters of the Alphabet, *Journal of Experimental Psychology: Human Perception and Performance*, **10**(5): 655–666.

Gonzalez, R. C. and Woods, R. E. (1992). *Digital Image Processing*, Addison Wesley, Reading, MA. ISBN 0-201-50803-6.

Gregory, R. L. (1977). Vision with Isoluminant Colour Contrast: 1. a Projection Technique and Observations, *Perception*, **6**: 113–119.

Gregory, R. L. (1990). *Eye and Brain: the Psychology of Seeing*, fourth edn, Weidenfeld and Nicolson, London.

Gregory, R. L. (1993). Seeing by Exploring, *in* S. R. Ellis, M. K. Kaiser and A. J. Grunwald (eds), *Pictorial Communication in Virtual and Real Environments*, second edn, Taylor and Francis, chapter 21, pp. 328–337. ISBN 0-74840-0082-6.

Hallett, P. E. (1991). Some Limitations to Human Peripheral Vision, *in* J. R. Cronly-Dillon (ed.), *Vision and Visual Dysfunction: Limits of Vision*, Vol. 5, MacMillan Press, Ltd., chapter 6, pp. 44–80. ISBN 0-333-52713-5.

Hamann, B. (1994). A Data Reduction Scheme for Triangulated Surfaces, *Computer Aided Geometric Design*, **11**(2): 197–214.

Hardin, C. L. (1988). *Color For Philosophers*, Hackett Publishing Company, Indianapolis, CA.

Harvey, L. O. and Gervais, M. J. (1981). Internal Representation of Visual Texture as the Basis for the Judgement of Similarity, *Journal of Experimental Psychology: Human Perception Performance*, **7**(4): 741–753.

Hawkes, R., Rushton, S. and Smyth, M. (1995). Update Rates and Fidelity in Virtual Environments, *Virtual Reality: Research, Development and Application*, **1**(2): 99–108.

He, T., Hong, L., Kaufman, A., Varshney, A. and Wang, S. (1995). Voxel Based Object Simplification, *Proceedings of Visualization '95*, pp. 296–303.

Heeley, D. (1991). Spatial Frequency Difference Thresholds Depend on Stimulus Area, *Spatial Vision*, **5**(3): 205–217.

Held, R. and Durlach, N. (1993). Telepresence, Time Delay and Adaption, *in* S. R. Ellis, M. K. Kaiser and A. J. Grunwald (eds), *Pictorial Communication in Virtual and Real Environments*, second edn, Taylor and Francis, chapter 14, pp. 232–246. ISBN 0-74840-0082-6.

Helman, J. (1994). Designing Real-Time Graphics for Entertainment, *Architecture and Performance of Entertainment Systems*, Vol. 14 of *SIGGRAPH '94 Lecture Notes*, ACM Press, chapter 1.

Hinker, P. and Hansen, C. (1993). Geometric Optimization, *Proceedings of Visualization '93*, pp. 189–195.

Hitchner, L. E. and McGreevy, M. W. (1993). Methods for User-Based Reduction of Model Complexity for Virtual Planetary Exploration, *Proceedings of the SPIE – The International Society for Optical Engineering*, Vol. 1913, pp. 622–36.

Holloway, R. L. (1991). Viper: a Quasi-Real-Time Virtual-Worlds Application, *UNC Technical Report No. TR-92-004*, Deptartment of Computer Science, University of North Carolina, Chapel Hill, NC.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. (1992). Surface Reconstruction from Unorganised Points, *Computer Graphics (SIGGRAPH'92 Proceedings)*, **26**(2): 71–78.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. (1993). Mesh Optimization, *Computer Graphics Proceedings, Annual Conference Series*, pp. 19–25.

Hubel, D. and Wiesel, T. (1962). Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat's Visual Cortex, *Journal of Physiology*, **160**: 106–154.

Humphreys, G. W. and Bruce, V. (1991). *Visual Cognition: Computational, Experimental and Neuropsychological Perspectives*, Hove Lawrence Erlbaum Associates. ISBN 0-863-77125-4.

Hurault, F. (1993). A Head Slaved Visual System for Flight Simulators, *Proceedings of the International Training Equipment Conference and Exhibition*, London, UK, pp. 37–42.

Hurvich, L. M. (1981). *Color Vision*, Sinauer Associates, Sunderland, Mass.

Hwang, S. C. and Yang, H. S. (1993). Efficient View Sphere Tessellation Method Based on Halfedge Data Structure and Quadtree, *Computers and Graphics*, **17**(5): 575–581.

Jacobson, R. E. (1995). Image Quality Metrics, *The Journal of Photographic Science*, **43**(2): 42–43.

Kalawsky, R. S. (1993). *The Science of Virtual Reality and Virtual Environments*, Addison-Wesley, Reading, MA. ISBN 0-201-63171-7.

Kalvin, A. D. and Taylor, R. H. (1994). Superfaces: Polygonal Mesh Simplification with Bounded Error, *Technical Report RC 19808 (#87702)*, IBM Research Division, T. J. Watson Research Centre, Yorktown Heights, NY.

Katz, D. (1951). *Gestalt Psychology*, Methuen and Co. Ltd., London.

Kelly, D. H. (1975). Spatial Frequency Selectivity in the Retina, *Vision Research*, **15**: 665–672.

Kelly, D. H. (1979). Motion and Vision. II. Stabilized Spatio-Temporal Threshold Surface, *Journal of the Optical Society of America*, **69**(10): 1340–1349.

Kelly, D. H. (1984). Retinal Inhomogenity: I. Spatiotemporal Contrast Sensitivity, *Journal of the Optical Society of America A*, **1**(1): 107–113.

Kemeny, A. (1993). A Cooperative Driving Simulator, *Proceedings of the International Training Equipment Conference (ITEC)*, London, UK, pp. 67–71.

Khan, G. N. and Giles, D. F. (1992). Extracting Contours by Perceptual Grouping, *Image and Vision Computing*, **10**(2): 77–88.

Kingslake, R. (1991). *An Introductory Course in Computer Graphics*, second edn, Chartwell-Bratt. ISBN 0-86238-284-X.

Koenderink, J. J., Bouman, M. A., de Mesquita, A. E. B. and Slappendel, S. (1978a). Perimetry of Contrast Detection Thresholds of Moving Spatial Sine Wave Patterns. II. The Far Peripheral Visual Field (Eccentricity 0°–50°), *Journal of the Optical Society of America*, **68**(6): 850–854.

Koenderink, J. J., Bouman, M. A., de Mesquita, A. E. B. and Slappendel, S. (1978b). Perimetry of Contrast Detection Thresholds of Moving Spatial Sine Wave Patterns. I. The Near Peripheral Visual Field (Eccentricity 0°–8°), *Journal of the Optical Society of America*, **68**(6): 845–849.

Korein, J. and Badler, N. (1983). Temporal Anti-Aliasing in Computer Generated Animation, *Computer Graphics*, **17**(3): 377–388.

Lamming, D. (1991a). On the Limits of Visual Detection, *in* J. R. Cronly-Dillon (ed.), *Vision and Visual Dysfunction: Limits of Vision*, Vol. 5, MacMillan Press, Ltd., chapter 2, pp. 6–14. ISBN 0-333-52713-5.

Lamming, D. (1991b). Signal Detection Theory, *in* J. R. Cronly-Dillon (ed.), *Vision and Visual Dysfunction: Limits of Vision*, Vol. 5, MacMillan Press, Ltd., chapter 3, pp. 15–22. ISBN 0-333-52713-5.

Lamming, D. (1991c). Spatial Frequency Channels, *in* J. R. Cronly-Dillon (ed.), *Vision and Visual Dysfunction: Limits of Vision*, Vol. 5, MacMillan Press, Ltd., chapter 8, pp. 97–105. ISBN 0-333-52713-5.

Levi, D. M., Klein, S. A. and Aitsebaomo, A. P. (1985). Vernier Acuity, Crowding and Cortical Magnification, *Vision Research*, **25**: 963–971.

Levitt, H. (1971). Transformed Up–Down Methods in Psychoacoustics, *The Journal of the Acoustical Society of America*, **49**(2): 467–477.

Levoy, M. and Whitaker, R. (1990). Gaze–Directed Volume Rendering, *ACM SIGGRAPH Special Issue on 1990 Symposium on Interactive 3D Graphics*, **24**(2): 217–223.

Liang, J., Shaw, C. and Green, M. (1991). On Temporal-Spatial Realism in the Virtual Reality Environment, *Proceedings of the 4th Annual Symposium on User Interface Software and Technology*, Hilton Head SC, pp. 19–25.

Lindstrom, P., Koller, D., Hodges, L. F., Ribarsky, W., Faust, N. and Turner, G. (1995). Level-of-Detail Management for Real-Time Rendering of Phototextured Terrain, *Technical Report No. TR95-06*, Graphics, Visualization and Usability Centre, Georgia Institute of Technology, Altanta, GA.

Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L. F., Faust, N. and Turner, G. (1996). Real-Time, Continuous Level of Detail Rendering of Height Fields, *Technical Report No. TR96-02*, Graphics, Visualization and Usability Centre, Georgia Institute of Technology, Altanta, GA.

Livingstone, M. S. (1988). Art, Illusion and the Visual System, *Scientific American*, **258**(1): 68–75.

Livingstone, M. S. and Hubel, D. H. (1988). Segregation of Form, Color, Movement, and Depth: Anatomy, Physiology, and Perception, *Science*, **240**: 740–749.

Lorensen, W. E. and Cline, H. E. (1987). Marching Cubes: a High Resolution 3D Surface Construction Algorithm, *Computer Graphics (SIGGRAPH '87 Proceedings)*, Vol. 21, pp. 163–169.

Lounsbery, M., DeRose, T. and Warren, J. (1994). Multiresolution Analysis for Surfaces of Arbitrary Topological Type, *Technical Report No. 93-10-05b*, Deptartment of Computer Science and Engineering, University of Washington.

Luebke, D. (1996). Hierarchical Structures for Dynamic Polygonal Simplification, *UNC Technical Report No. TR96-006*, Deptartment of Computer Science, University of North Carolina, Chapel Hill, NC.

Luo, M. R. and Rigg, B. (1987). BFD(l:c) Colour-Difference Formula, Part 1: Development of the Formula, *Journal of the Society of Dyers and Colourists*, **103**(2): 86–94.

MacAdam, D. L. (1985). Evaluation of Color Differences, *Journal of the Optical Society of America A (Optics and Image Science)*, **2**(13): 11.

MacDonald, L. W., Luo, M. R. and Scrivener, S. A. R. (1990). Factors Affecting the Appearance of Coloured Images on a Video Display Monitor, *Journal of Photographic Science*, **38**(4–5): 177–186.

Maciel, P. and Shirley, P. (1995). Visual Navigation of Large Environments Using Textured Clusters, *Symposium on Interactive 3D Graphics*, pp. 95–102.

Mannos, J. L. and Sakrison, D. J. (1974). The Effects of a Visual Fidelity Criterion on the Encoding of Images, *IEEE Transactions on Information Theory*, **IT-20**(4): 525–536.

Marr, D. (1982). *Vision: a Computational Investigation into the Human Representation and Processing of Visual Information*, W. H. Freeman, New York, NY.

DeHaemer, Jr., M. J. and Zyda, M. J. (1991). Simplification of Objects Rendered by Polygonal Approximations, *Computer and Graphics*, **15**(2): 175–184. Pergamon Press, UK.

VR News (1995). Headmounted displays, *VR News*, **4**(4): 20–29.

McMillan, L. and Bishop, G. (1995). Plenoptic Modeling: An Image-Based Rendering System, *in* R. Cook (ed.), *SIGGRAPH '95 Conference Proceedings*, pp. 39–46.

Meyer, G. W. and Greenberg, D. P. (1980). Perceptual Color Spaces for Computer Graphics, *Computer Graphics*, **14**: 254–261.

Money, K. (1970). Motion Sickness, *Physiological Reviews*, **50**(1): 1–39.

Morgan, M. J. (1991). Hyperacuity, *in* D. Regan (ed.), *Vision and Visual Dysfunction: Spatial Vision*, Vol. 10, MacMillan Press, Ltd., chapter 4, pp. 87–110. ISBN 0-333-52713-5.

Mullen, K. T. (1985). The Contrast Sensitivity of Human Color Vision to Red-Green and Blue-Yellow Chromatic Gratings, *Journal of Physiology*, **359**: 381–400.

Mullen, K. T. and Kingdom, F. A. A. (1991). Colour Contrast in Form Perception, *in* J. Cronly-Dillon (ed.), *Vision and Visual Dysfunction: The Perception of Colour*, Vol. 6, MacMillan Press, Ltd., chapter 12, pp. 198–217.

MultiGen, Inc. (1994). *ModelGen2 Modeler's Guide*, MultiGen Inc., San Jose, CA. Revision 14.1.

Murphy, B. J. (1978). Pattern Thresholds for Moving and Stationary Gratings During Smooth Eye Movement, *Vision Research*, **18**: 521–530.

Nachmias, J. (1968). Visual Resolution of Two-Bar Patterns and Square-Wave Gratings, *Journal of the Optical Society of America*, **58**(1): 9–13.

Nakayama, K. (1990). Properties of Early Motion Processing: Implications for the Sensing of Egomotion, *in* R. Warren and A. H. Wertheim (eds), *The Perception and Control of Self Motion*, Lawrence Erlbaum, Hillsdale, NJ, pp. 69–80.

NSF (1992). Research Directions in Virtual Environments: Report of an NSF Invitational Workshop, *Computer Graphics*, **26**(3): 153–177.

Ohshima, T., Yamamoto, H. and Tamura, H. (1996). Gaze-Directed Adaptive Rendering for Interacting with Virtual Space, *Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS)*, Santa Clara, CA, pp. 103–110.

Oman, C. M. (1993). Sensory Conflict in Motion Sickness: an Observer Theory approach, *in* S. R. Ellis, M. K. Kaiser and A. J. Grunwald (eds), *Pictorial Communication in Virtual and Real Environments*, second edn, Taylor and Francis, chapter 24, pp. 362–376. ISBN 0-74840-0082-6.

Owsley, C. J., Sekuler, R. and Siemsen, D. (1983). Contrast Sensitivity Throughout Adulthood, *Vision Research*, **23**: 689–699.

Peli, E. (1990). Contrast in Complex Images, *Journal of the Optical Society of America A (Optics and Image Science)*, **7**(10): 2032–2040.

Phillips, P. L. (1986). Minimum Colour Differences Required to Recognize Small Objects on a Colour CRT, *Journal of the Institution of Electronic and Radio Engineers*, **56**(3): 123–129.

Pokorny, J., Shevell, S. K. and Smith, V. C. (1991). Colour Appearance and Colour Constancy, *in* J. Cronly-Dillon (ed.), *Vision and Visual Dysfunction: The Perception of Colour*, Vol. 6, MacMillan Press, Ltd., chapter 4, pp. 43–61.

Potmesil, M. and Chakravarty, I. (1983). Modeling Motion Blur in Computer-Generated Images, *Computer Graphics*, **17**(3): 389–399.

Poynton, C. A. (1993). "Gamma" and its Disguises: The Nonlinear Mappings of Intensity in Perception, CRTs, Film and Video, *SMPTE Journal (Society of Motion Picture and Television Engineers)*, **102**(12): 1099–1108.

Reddy, M. (1994). Reducing Lags in Virtual Reality Systems using Motion-Sensitive Level of Detail, *Proceedings of the 2nd UK VR-SIG Conference*, Theale, Reading, UK, pp. 25–31.

Reddy, M. (1995a). A Perceptual Framework for Optimising Visual Detail in Virtual Environments, *Proceedings of the FIVE '95 Conference*, QMW, University of London, UK, pp. 189–201.

Reddy, M. (1995b). A Survey of Level of Detail Support in Current Virtual Reality Solutions, *Virtual Reality: Research, Development and Application*, **1**(2): 95–98.

Reddy, M. (1995c). Musings on Volumetric Level of Detail for Virtual Environments, *Virtual Reality: Research, Development and Application*, **1**(1): 49–56.

Reddy, M. (1996a). A Measure for Perceived Detail in Computer-Generated Images, *Technical Report ECS-CSG-19-96*, Department of Computer Science, University of Edinburgh.

Reddy, M. (1996b). SCROOGE: Perceptually-Driven Polygon Reduction, *Computer Graphics Forum*, **15**(4): 191–203.

Reddy, M. (1997). The Development and Evaluation of a Model of Visual Acuity for Computer-Generated Imagery, *Technical Report ECS-CSG-30-97*, Department of Computer Science, University of Edinburgh.

Reed, T. R. and du Buf, J. M. H. (1993). A Review of Recent Texture Segmentation and Feature Extraction Techniques, *Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding*, **57**(3): 359–372.

Reed, T. R. and Wechsler, H. (1990). Segmentation of Textured Images and Gestalt Organisation Using Spatial/Spatial-Frequency Representations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(1): 1–12.

Regan, C. (1995). An Investigation into Nausea and Other Side-effects of Head-coupled Immersive Virtual Reality, *Virtual Reality: Research, Development, Applications*, **1**(1): 17–32.

Regan, D. and Beverley, K. I. (1983). Visual Field Described by Contrast Sensitivity, by Acuity and by Relative Sensitivity to Different Orientations, *Investigative Ophthalmology and Visual Science,* **24**: 754–759.

Renka, R. J. (1984). Interpolation of Data on the Surface of a Sphere, *ACM Transactions on Mathematical Software*, **10**: 417–436.

Rieger, J. H. and Toet, L. (1985). Human Visual Navigation in the Presence of 3-D Motions, *Biological Cybernetics*, **52**: 354–360.

Ritter, J. (1990). A Fast Approximation to 3D Euclidean Distance, *in* A. S. Glassner (ed.), *Graphics Gems*, Vol. 1, Academic Press, inc, pp. 432–433. ISBN 0-12-286165-5.

Robinson, D. A. (1964). The Mechanics of Human Saccadic Eye Movements, *Journal of Physiology*, **180**: 569–590.

Roehl, B. (1995). Personal communication.

Rohlf, J. and Helman, J. (1994). IRIS Performer: a High Performance Multiprocessing Toolkit for Real-Time 3D Graphics, *Computer Graphics (SIGGRAPH '94 Proceedings)*, pp. 381–394.

Rossignac, J. R. and Borrel, P. (1992). Multi-Resolution 3D Approximations for Rendering Complex Scenes, *Technical Report RC 17697 (#77951)*, IBM Research Division, T. J. Watson Research Centre, Yorktown Heights, NY.

Rovamo, J. and Virsu, V. (1979). An Estimation and Application of the Human Cortical Magnification Factor, *Experimental Brain Research*, **37**: 495–510.

Rushmeier, H., Ward, G., Piatko, C., Sanders, P. and Rust, B. (1995). Comparing Real and Synthetic Images: Some Ideas About Metrics, *Proceedings of the 6th Eurographics Workshop on Rendering*, Dublin, pp. 82–91.

Scarlatos, L. L. (1990). A Refined Triangulation Hierarchy for Multiple Levels of Terrain Detail, *Proceedings of the IMAGE V Conference*, Phoenix, Arizona, pp. 115–122.

Schachter, B. J. (1981). Computer Image Generation for Flight Simulation, *IEEE Computer Graphics and Applications*, **1**: 29–68.

Schade, O. H. (1956). Optical and Photoelectric Analog of the Eye, *Journal of the Optical Society of America*, **46**: 721–739.

Schmitt, F. J. M., Barsky, B. A. and Du, W.-H. (1986). An Adaptive Subdivision Method for Surface-Fitting from Sample Data, *Computer Graphics*, **20**(4): 179–188.

Schor, C. M. and Badcock, D. R. (1985). A Comparison of Stereo and Vernier Acuity with Spatial Channels as a Function of Distance from Fixation, *Vision Research*, **25**: 1113–1119.

Schroeder, W. J., Zarge, J. A. and Lorensen, W. E. (1992). Decimation of Triangle Meshes, *Computer Graphics (SIGGRAPH '92 Proceedings)*, Vol. 26, pp. 65–70.

Sekuler, R. and Blake, R. (1994). *Perception*, third edn, McGraw-Hill, Inc., New York, NY. ISBN 0-07-113683-5.

Sen, R., Yates, R. B. and Thacker, N. A. (1995). Virtual Reality Based on Cost/Benefit Analysis, *Proceedings of the FIVE '95 Conference*, QMW, University of London, UK, pp. 213–221.

Sewell, J. (1995). Automatic Generation of Simple Replacements for Groups of Objects. The University of Cambridge Computer Laboratory, UK.

Shaw, C., Liang, J., Green, M. and Sun, Y. (1992). The Decoupled Simulation Model for Virtual Reality Systems, *Proceedings of the CHI'92*, pp. 321–328.

Sheridan, T. B. (1992). Musings on Telepresence and Virtual Presence, *Presence: Teleoperators and Virtual Environments*, **1**(1): 120–126.

Silverstein, L. D. and Merrifield, R. M. (1985). The Development and Evaluation of Color Systems for Airborne Applications, *Technical Report DOT/FAA/PM–85–19*, Federal Aviation Administration, Washington, DC.

Simpson, W. A. (1989). The Step Method: A New Adaptive Psychophysical Procedure, *Perceptual Psychology*, **45**: 572–576.

Singhal, S. K. and Cheriton, D. R. (1995). Exploiting Position History for Efficient Remote Rendering in Networked Virtual Reality, *Presence: Teleoperators and Virtual Environments*, **4**(2): 169–193.

Slater, M. and Usoh, M. (1993). Presence in Immersive Virtual Environments, *Proceedings of VRAIS '93 (IEEE Virtual Reality International Symposium)*, Seattle, Washington, pp. 90–96.

Slater, M., Usoh, M. and Chrysanthou, Y. (1995). The Influence of Dynamic Shadows on Presence in Immersive Virtual Environments. Department of Computer Science, Queen Mary Westfield College, University of London.

Smets, G. J. and Overbeeke, K. J. (1995). Trade-Off Between Resolution and Interactivity in Spatial Task Performance, *IEEE Computer Graphics and Applications*, **15**(5): 46–51.

Soufi, B. and Scrivener, S. A. R. (1992). Perceptual Grouping Algorithms and Object Identification, *Technical Report 92/C/LUTCHI/0148*, Loughborough University of Technology, LUTCHI Research Centre.

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

Stollnitz, E. J., DeRose, T. D. and Salesin, D. H. (1995a). Wavelets for Computer Graphics: a Primer, Part 1, *IEEE Computer Graphics and Applications*, **15**(3): 76–84.

Stollnitz, E. J., DeRose, T. D. and Salesin, D. H. (1995b). Wavelets for Computer Graphics: a Primer, Part 2, *IEEE Computer Graphics and Applications*, **15**(4): 75–85.

Stork, D. G. and Wilson, H. R. (1990). Do Gabor Functions Provide Appropriate Descriptions of Visual Cortical Receptive Fields?, *Journal of the Optical Society of America A*, **7**(8): 1362–1373.

Sutherland, I. E. (1965). The Ultimate Display, *Proceedings of the IFIP Congress*, pp. 506–508.

Sutter, E. E. and Tran, D. (1991). The Field Topography of ERG Components in Man—I. The Photopic Luminance Response, *Vision Reseach*, **32**(3): 433–446.

Swartz, M., Wallace, D. and Tkacz, S. (1992). The Influence of Frame Rate and Resolution Reduction on Human Performance, *Proceedings of the Human Factors Society 36th Annual Meeting*, Santa Monica, CA, pp. 1440–1444.

Tipton, D. A. (1984). A Review of Vision Physiology, *Aviation, Space and Environmental Medicine*, **55**(2): 145–149.

Travis, D. (1991). *Effective Color Displays: Theory and Practice*, Academic Press Ltd., London, UK. ISBN 0–12-697690-2.

Turk, G. (1992). Re-tiling Polygonal Surfaces, *in* E. E. Catmull (ed.), *Computer Graphics (SIGGRAPH ’92 Proceedings)*, Vol. 26, pp. 55–64.

Tyler, C. W. (1985). Analysis of Visual Modulation Sensitivity. II. Peripheral Retina and the Role of Photoreceptor Dimensions, *Journal of the Optical Society of America*, **A2**(3): 393–398.

Uliano, K., Kennedy, R. and Lambert, E. (1986). Asynchronous Visual Delays and the Development of Simulator Sickness, *Proceedings of the Human Factors Society (30th Annual Meeting)*, Human Factors Society, Dayton, OH, pp. 422–426.

van Dam, A. (1993). Virtual Reality as a Forcing Function: Software Implications of a New Paradigm, *IEEE Symposium on Research Frontiers in Virtual Reality*.

Varshney, A. (1994). *Hierarchical Geometric Approximations*, PhD thesis, Department of Computer Science, University of North Carolina at Chapel Hill.

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

Virsu, V., Rovamo, J., Laurinen, P. and Näsänen, R. (1982). Temporal Contrast Sensitivity and Cortical Magnification, *Vision Research*, **33**: 1211–1217.

Wann, J. P., Rushton, S. K. and Lee, D. N. (1995). Can You Control Where You Are Heading When You Are Looking At Where You Want To Go?, *in* B. Bardy, R. Bootsma and Y. Guiard (eds), *Proceedings of the 8th International Conference on Event Perception and Action*, pp. 171–174.

Ware, C. and Knight, W. (1995). Using Visual Texture for Information Display, *ACM Transactions on Graphics*, **14**(1): 3–20.

Warren, W. H. and Hannon, D. J. (1990). Eye Movements and Optical Flow, *Journal of the Optical Society of America*, **A**(7): 160–169.

Watson, A. B. (1983). Detection and Recognition of Simple Spatial Forms, *in* O. J. Braddick and A. C. Sleigh (eds), *Physical and Biological Processing of Images*, Springer-Verlag, New York, pp. 100–114.

Watson, A. B. (1987). The Cortex Transform: Rapid Computation of Simulated Neural Images, *Computer Vision, Graphics, and Image Processing (CVGIP)*, **39**: 311–327.

Watson, B., Walker, N. and Hodges, L. F. (1995). A User Study Evaluating Level of Detail Degradation in the Periphery of Head-Mounted Displays, *Proceedings of the FIVE '95 Conference*, QMW, University of London, UK, pp. 203–212.

Wernecke, J. (1993). *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor(TM), Release 2*, Addison-Wesley, Reading, MA. ISBN 0-201-62495-8.

Wetherill, G. B. and Levitt, H. (1965). Sequential Estimation of Points on a Psychometric Function, *The British Journal of Mathematical and Statistical Psychology*, **18**(1): 1–10.

Williams, M. (1993). Immersive Virtual Environments – Some Implications for Computer Graphics, *Report on the One Day Immersive Virtual Environments Workshop*, Queen Mary and Westfield College, London.

Wilson, H. R. and Bergen, J. R. (1979). A Four Mechanism Model for Threshold Spatial Vision, *Vision Research*, **19**: 19–32.

Witmer, B. G. and Singer, M. J. (1994). Measuring Presence in Virtual Environments, *Technical Report 1014*, U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, VA.

Wloka, M. M. (1993a). Incorporating Update Rates into Today's Graphics Systems, *Technical Report CS-93-56*, Department of Computer Science, Brown University, Providence, RI.

Wloka, M. M. (1993b). Thesis Proposal: Time-Critical Graphics, *Technical Report CS-93-50*, Department of Computer Science, Brown University, Providence, RI.

Wloka, M. M. (1995). Lag in Multiprocessor Virtual Reality, *Presence: Teleoperators and Virtual Environments*, **4**(1): 50–63.

Wloka, M. M., Zeleznik, R. C. and Miller, T. (1995). Practically Frameless Rendering, *Technical Report CS-95-06*, Department of Computer Science, Brown University, Providence, RI.

Wyszecki, G. and Fielder, G. H. (1971). New Color-Matching Ellipses, *Journal of the Optical Society of America*, **61**(9): 1135–1152.

Yan, J. K. (1985). Advances in Computer Generated Imagery for Flight Simulation, *IEEE Computer Graphics and Applications*, **5**: 37–51.

Young, L. R. and Sheena, D. (1975). Survey of Eye Movement Recording Methods, *Behaviour Research Methods and Instrumentation*, **7**(5): 397–429.

Zeki, S. (1993). *A Vision of the Brain*, Blackwell Scientific Publications, Inc., Oxford, UK. ISBN 0-632-03054-2.