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.

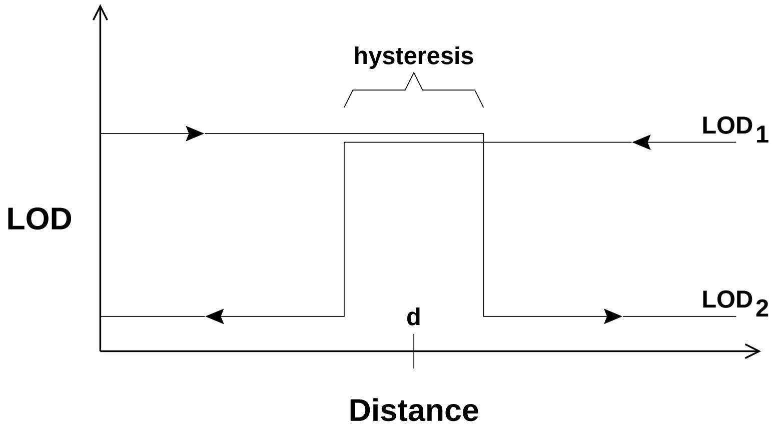

Figure 2.1: *Distance-based switching thresholds between two LODs illustrating a period of hysteresis around the threshold distance, d. Adapted from Astheimer and Pöche (1994).*

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. **Proximity** : 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-

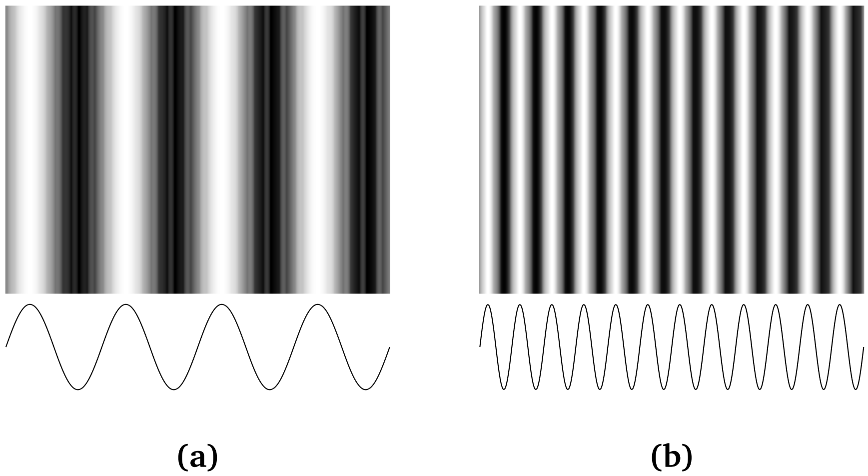

Figure 2.2: *An illustration of two contrast gratings displaying: (a) a low and, (b) a higher spatial frequency. The curve below each of the gratings shows the sinusoidal nature of the intensity distribution. If grating (a) was positioned to occupy 1 deg of visual arc, then it would have a spatial frequency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

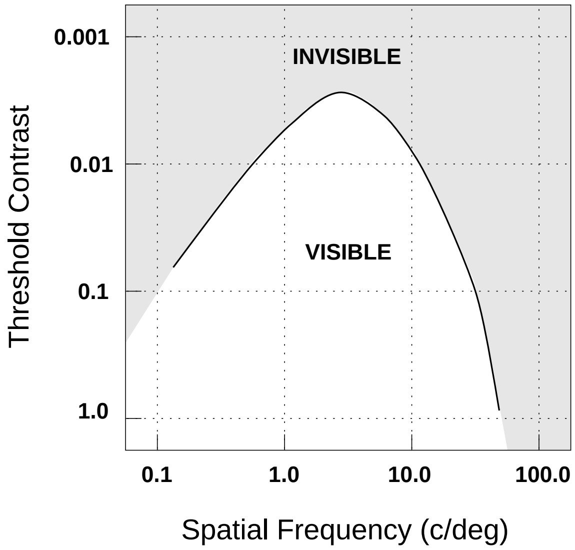

Figure 2.3: *An example contrast sensitivity function for static detail. This represents the size-sensitivity of the human visual system in relation to the contrast of the stimulus.*

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

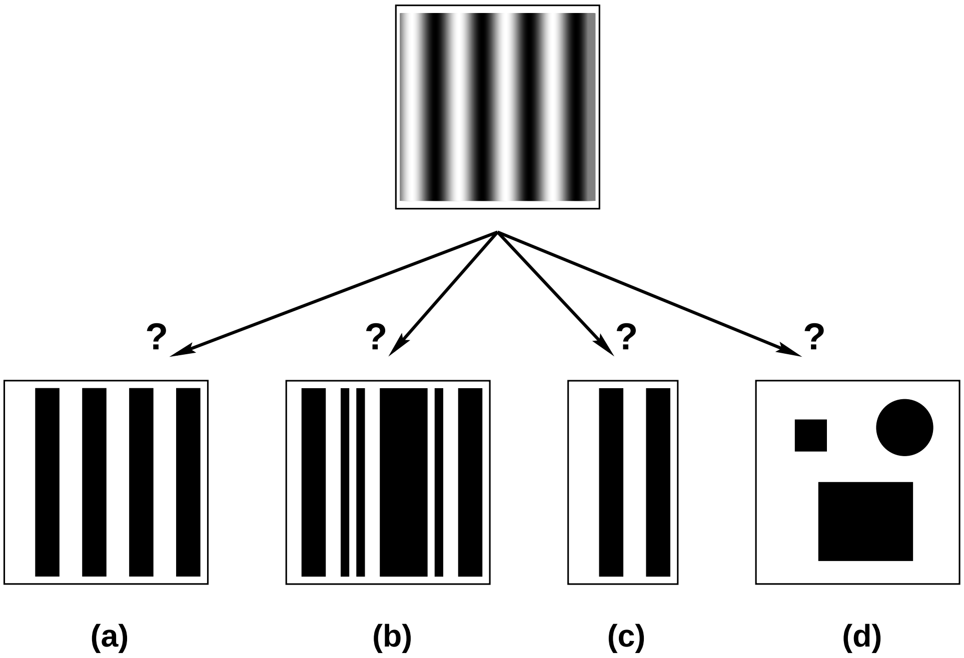

Figure 2.4: *How does a simple 1D harmonic contrast grating relate to: a) square-wave gratings, b) complex gratings, c) gratings of different periodicity, and d) 2D gratings?.*

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,

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

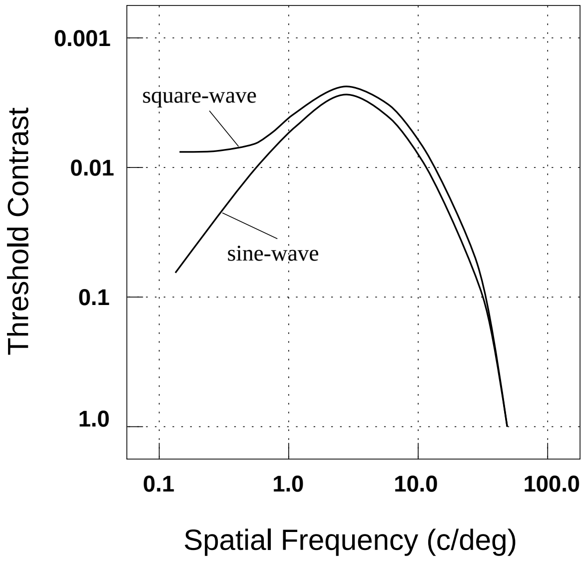

Figure 2.5: *Contrast sensitivity compared for sine-wave gratings (lower curve) and square-wave gratings (upper curve). Adapted from results in Campbell and Robson (1968).*

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.

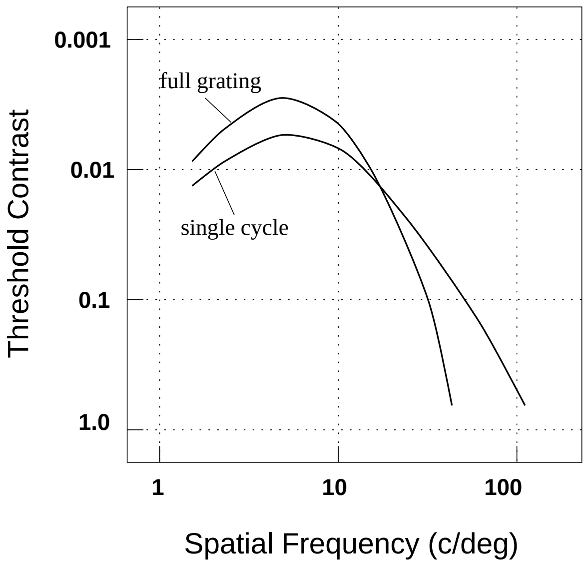

Figure 2.6: *Contrast sensitivity compared for full gratings and single cycle (double bar) gratings. Adapted from Campbell et al. (1969).*

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.

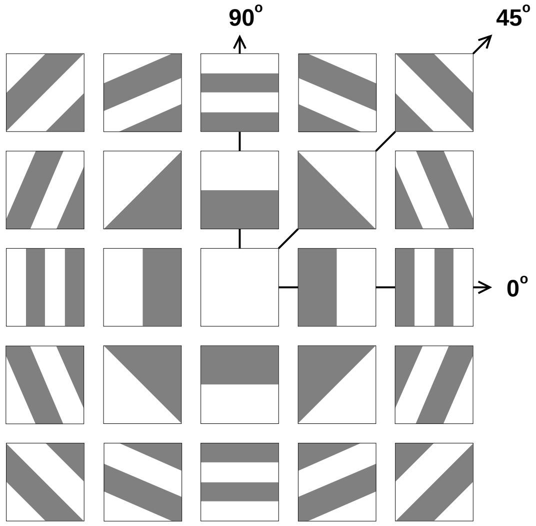

Figure 2.7: *An illustration of 1 cycle and 2 cycle spatial frequencies at various orientations.*

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

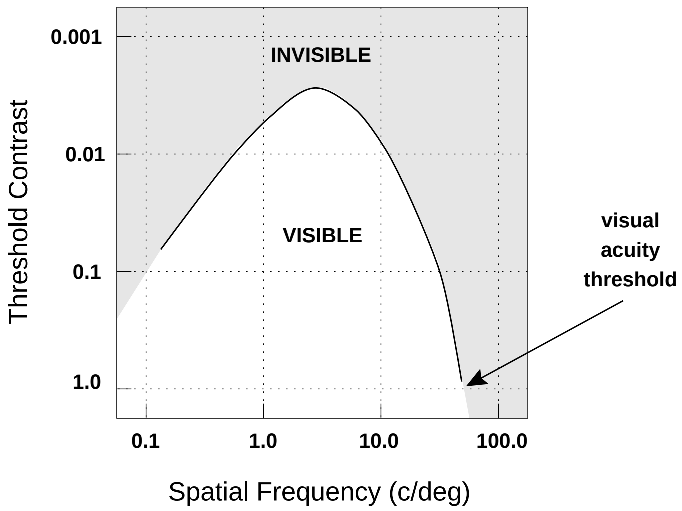

Figure 2.8: *A portrayal of how visual acuity can be described in terms of spatial frequency given a subject's contrast sensitivity function.*

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.