Eye-movements and Interactive Graphics.

Carol O'Sullivan, John Dingliana, Sarah Howlett
Image Synthesis Group
Trinity College Dublin
Dublin 2
Ireland
E-Mail: Carol.OSullivan,John.Dingliana@cs.tcd.ie

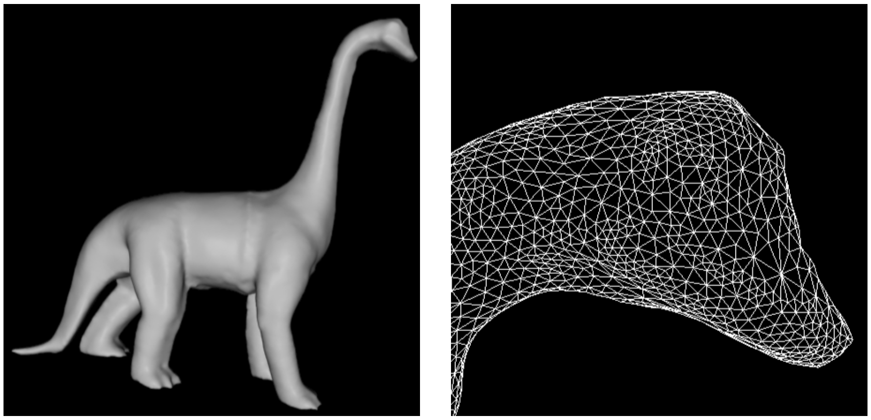
July 1, 2002

Abstract

In this chapter, we will discuss the usefulness of eye-tracking for computer applications that attempt to render simulations at interactive rates. Some recent research that uses this technology is presented. While eye-tracking has been relatively well-explored for the purposes of accelerating real-time rendering, it has not been very much considered with respect to the actual simulation process. In fact, perceptual issues relating to physically-based simulation have been largely neglected in the field of computer graphics. To evaluate simulations, we explore the relationship between eye-movements and dynamic events, such as collisions. We will describe some gaze-contingent systems that we and others have developed that exploit eye-tracking to enhance the perceived realism of images and simulations. Some experiments that were carried out to evaluate the feasibility of this approach are also presented.

1 Introduction

Researchers in the field of computer graphics are concerned with producing images and animations that are as realistic as possible, mainly to human viewers. Creating images consists of two phases: *modelling* a scene containing objects and environmental effects and storing an appropriate representation in digital format; and *rendering*, or displaying, these scenes using a variety of platforms and technologies. The two problems are not independent - the quality of the final image depends greatly on the accuracy of the models, as does the speed at which the scene may be rendered. The more complex and detailed the model, the longer it will take to produce a graphical representation of it. If we now consider the problem of producing an animation, we can see that both the above phases will form an integral part of this process, as each frame of an animation consists of a computer generated image of a digitally-represented model. At



A dinosaur model, consisting of 47904 triangles and 23984 vertices.

A detailed view of the triangles forming the head.

Figure 1: Objects are often modelled using many polygons.

each time-step of the animation, the positions and orientations of the models must be updated by some means, known as *simulation*, and an image rendered of the updated scene. When this animation is shown to a viewer, at the very least 10 frames per second must be displayed, or else the animation will appear to be very slow and jerky. In fact, most movie animations are generated at around 25 frames per second, while with interactive applications such as games the ideal frame rate is often much higher than this. In the case of a movie, the frames of the animation can be created in advance and then played back in real-time to the viewer. Hence, the speed at which the frames are rendered is not critical. However, for interactive animations each frame must be generated while the viewer watches, thus providing a formidable challenge for the graphics engine. Strategies that reduce this computational burden often produce poor-quality images and motions as a trade-off, but by taking perceptual factors into account and integrating eye-movement analysis, adaptive systems can be developed which improve the perceived quality of the degraded graphics.

2 Perception, Eye-movements and Graphics

In this section we will consider in more detail the phases of modelling objects with a limited amount of data, rendering these objects quickly and simulating their motions realistically, in each case examining how eye-movement analysis can be used to understand and improve performance and quality.

2.1 Modelling

In time-critical computer graphics applications, such as Virtual Reality (VR), three-dimensional objects are often represented as meshes of polygons (see Figure 1). The requirement in interactive systems for real-time frame rates means that a limited number of polygons can be displayed by the graphics engine in each frame of a simulation. Therefore, meshes with a high polygon count often have to be simplified in order to achieve acceptable display rates. The number of polygons and hence the *Level Of Detail (LOD)* of the model needs to be reduced (see Figure 2).

This can be achieved in two ways: a representation of the object at several levels of detail with a fixed polygon count can be generated, although switching between such levels of detail can cause a perceivable ”pop” in an animation; or special meshes, called multi-resolution or progressive meshes, can be built that can be refined at run-time, i.e. parts of an object may be refined or simplified based on the current view, thus allowing a smooth transition from lower levels to higher levels of detail. Many techniques exist for producing both types of simplification, but they usually choose the areas of the object to simplify based on properties of the surface, such as curvature, color or texture. For example, the popular QSlim simplification software, described in (Garland, 1999), is based on such principles. Several techniques exist which use perceptual principles to choose which objects or parts of objects to simplify at run-time, as discussed in Section 2.2.2, but perceptual factors are not considered while actually building the levels of detail. Simplifying objects based on surface properties alone does improve the visual quality of the resulting mesh significantly, with fewer jagged edges and facets visible. However, if the *semantics* of the object are not considered when simplifying, at lower levels of detail the object may become unrecognisable sooner than is necessary. For example, the ears of a bunny are the most important features that make it recognisable. But how can such semantics of be captured in some objective and measurable way?

When building level of detail meshes, we need to determine which triangles should be retained at highest detail for longest. Eye-tracking can be used to provide an objective measure of a viewer’s interest in a specific region. The IPoMM software (Interactive Perception of Multiresolution Meshes), described in (Janott, 1999; Janott and O’Sullivan, 1999), uses an interest-dependent strategy for decreasing the LOD of a given object. We display models at the highest level of detail to a number of human viewers, rotating it in all directions to eliminate view-dependency, while simultaneously tracking their eye-movements. A counter associated with each triangle is incremented each time it is fixated by the viewer. The number of fixations is then interpreted as a measure of the salience of that triangle, and the order with which regions are coarsened is updated accordingly. Using this strategy, regions of the object with higher saliency are retained at high resolution for as long as possible, while less interesting regions are simplified earlier. In this way, the object remains recognisable for longer (See Figure 3). Although this system has been used to generate multiresolution mesh structures to date, it is equally applicable to the generation of fixed LOD meshes.

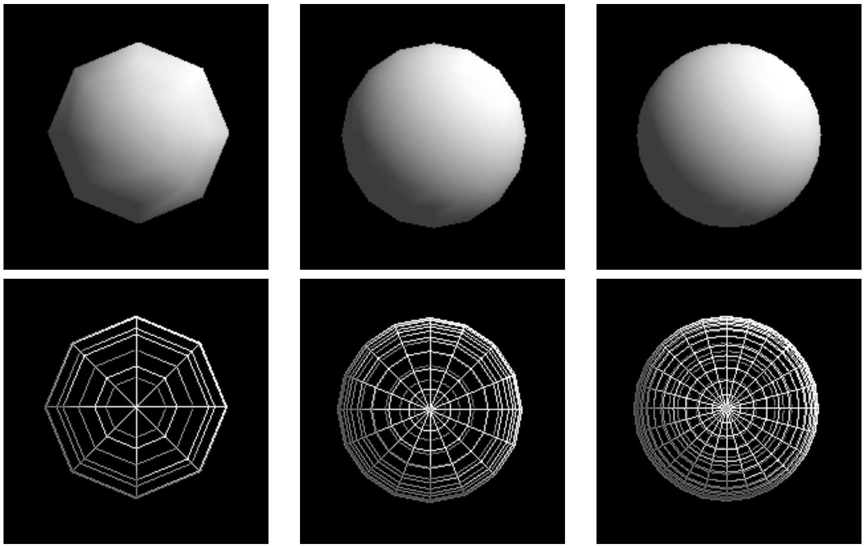
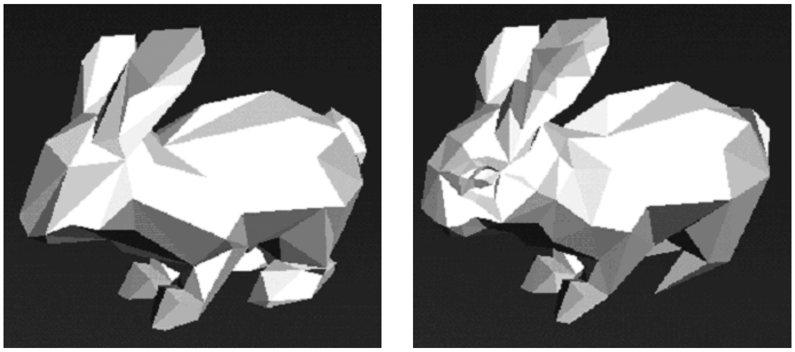


Figure 2: A sphere modelled at different levels of detail



358 polygons, simplification distributed over the whole object

358 polygons, high resolution at the head leaves less detail for the hind legs

Figure 3: IPoMM uses eye-tracking to determine the regions of interest

Figure 4: Fog is used to mask the popping up of trees in a game (eRacer ©Rage Software)

2.2 Rendering

We have discussed the importance of being able to reduce the Level of Detail (LOD) of models for interactive graphical systems, thus reducing the potential load on the graphics engine. This reduction in accuracy will also cause the visual quality of the graphics to be degraded in a number of ways. We have seen that preprocessing the models using eye-movement data can help to ameliorate this situation, but the situation can be further improved when the graphics engine is actually rendering the frames of an interactive animation. Adaptive systems use heuristics to determine which objects, or parts of objects, to refine at runtime. The system described in (Funkhouser and Séquin, 1993) was the first such system, which used factors such as the distance of an object from the viewer and its velocity to choose from a number of fixed LOD models to render.

The problems that arise with such an approach include popping effects that occur when the graphics engine suddenly switches from a lower to a higher level of detail. Nevertheless, this may be the method of choice for extremely time-critical systems, for example when the processing power is particularly limited and a small number of LOD models are stored for each simple object. Several visual tricks can be used to help mask the popping and we will discuss how real-time eye-movement analysis can help in Section 2.2.1. However, for more complex models, such as mountainous terrains, the costs of storing several versions of such a model would be prohibitive. If smooth, progressive switching between LODs is required, only parts of the object should be refined at runtime, based on heuristics as in the above-mentioned system. Gaze-contingent techniques, which incorporate eye-tracking or models of visual attention to direct this process, are discussed in Section 2.2.2.

2.2.1 Level of Detail popping and Change-Blindness

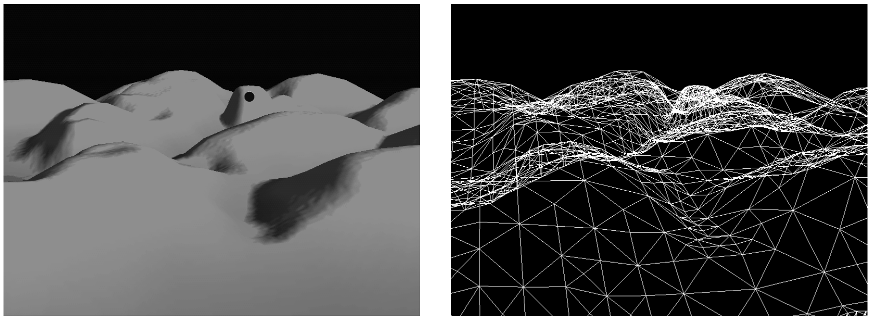
When processing power is limited, the graphics engine of an interactive system such as a game cannot render all objects that should be visible, nor can it render those objects at a consistently high level of detail. This means that whole objects must suddenly appear and disappear, as in Figure 4, or a lower or higher LOD for the object will suddenly be displayed, as discussed above. Tricks such as masking the changes using fog are possible, but these may not always be effective or appropriate. We propose that the phenomenon of *change blindness*, i.e. the inability of the brain to detect certain changes following saccades, could be a more effective means of hiding such events. For example, when a house suddenly pops up in a computer game, if this had happened during an eye-movement, the change should be less noticeable.

Many studies have shown that people are very slow to notice quite large changes made to an image during a saccade - see (Henderson and Hollingworth, 1999), (Rensink et al., 2000), (Simons, 2000), (Grimes, 1996), (O'Regan, 2001) and many others for a full discussion of this phenomenon. Saccades can either be detected using eye-tracking hardware - see (Triesch et al., 2002) for a system designed specifically for interactive graphics systems - or induced using the flicker paradigm (i.e. blanking the screen) or mud-splashes.

We wish to exploit change-blindness to mask LOD popping in an interactive graphics program such as a game. Therefore, we decided to first examine eye-movements while people play such games, our main aim being to establish the durations and frequency of saccades. This information is important if we want to develop a real-time system where changes occur during saccades. We did not find any previous study of eye-movements while playing games, but in (Andrews and Coppola, 1999), eye movements were recorded while viewers performed several tasks, including viewing a scene, watching repetitively moving patterns and performing an active search task. They recorded average fixation durations of .3, .6 and .22 seconds respectively for each of these tasks. To establish what patterns occur when playing an interactive graphical game, we conducted some preliminary experiments using the SMI EyeLink eye-tracker and recorded fixation and saccade durations. Eight subjects passively viewed video clips of a rollercoaster, and played two different types of game: a racing game, where the participant had to navigate a car around a racing track, and another where they had to shoot asteroids which could appear at any location on the screen.

The video clip provided the highest fixation duration average of .4 seconds, while the average fixation duration for the games was .22 and .25 seconds for the racing game and the asteroids respectively. Saccade duration averages were .038 and .045 seconds for the racing and asteroids games respectively. These results are consistent with those reported in (Andrews and Coppola, 1999), in that our passive task, i.e. viewing a video clip, had higher fixation durations than our active tasks, i.e. playing a game. Three of the participants were experienced games players and it was found that the average fixation duration for these players was higher for both games (by .126 and .08 seconds respectively). This is probably because they have less need to look around in order to effectively navigate or aim. The results found for the fixation duration when looking at the video clip was roughly the same as results found for other subjects. We also found that the saccade duration for experienced players is on average .09 seconds less than that for non-experienced players.

Using the results of these experiments, we are building an adaptive framework that allows objects to be added to a scene, or the fixed level of detail to



The terrain as it looks to the viewer - the black dod shows the fixation position

The terrain in wireframe mode, with increased detail where the viewer is looking

Figure 5: A multi-resolution mesh

be raised for objects that have increased in salience, during a saccade. Rather than allowing the popping to occur immediately when the viewer is at a fixed distance from the location of the object, once a particular threshold has been passed the object will then be popped during the next saccade. A list of such objects is maintained between subsequent saccades.

2.2.2 Gaze-Directed Rendering

In section 2.1, we saw how an object can be modelled to allow for simplification at run-time, and now we will discuss how the graphics engine will choose when and where to simplify such objects. (Funkhouser and Séquin, 1993) established the principle of exploiting the limitations of the human visual system to choose LODS for objects, by taking factors such as the velocity and size of the objects into account. (Reddy, 1997) was the first to examine in detail how a model of human visual acuity could guide simplification, determining when a change from one LOD to the next will be imperceptible to the viewer.

It is a well-known fact that people's ability to perform many types of visual processing tasks diminishes the more peripherally stimuli are presented (Aubert and Foerster, 1857; Weymouth, 1958). For a complete discussion of the physiological reasons for this eccentricity effect and the cortical magnification factor (M) which can in some circumstances be used to quantify it, see (DeValois and DeValois, 1988), (Rovamo and Virsu, 1979) and (Carrasco and Frieder, 1997). This effect has been investigated in the field of reading and scene perception also, see (Rayner, 1986; Shioiri and Ikeda, 1989; van Diepen et al., 1998) and others. Bearing this in mind, it is clear that simplifications to objects or regions of objects in the periphery of a viewer's visual field should be less perceptible than those close to their fixation point. Exploiting this effect while rendering objects at multiple LODS is known as *gaze-directed rendering.* Examples of systems that use this technique may be found in (Ohshima et al., 1996) and

(Murphy and Duchowski, 2001). However, eccentricity alone is often not sufficient to predict visual acuity. For example, in (Strasburger et al., 1994) both size and contrast had to be increased to neutralise the eccentricity effect. Considering eccentricity together with other perceptual factors when simplifying is therefore a more favourable approach. A method which considers eccentricity, contrast, spatial frequency and silhouette-preservation is described in (Luebke et al., 2000).

As discussed above, there are circumstances when choosing a fixed LOD for an object at runtime will not be desirable or feasible. Instead, a special mesh structure, called a multi-resolution or progressive mesh, can be used to avoid having to display a whole object at the highest level of detail even though only part of it may be currently important based on the user's view. The system described in (Luebke et al., 2000) used just such a view-dependent simplification technique. When an eye-tracker is used with such meshes, detail can be added wherever the user is looking. This will give the impression that the mesh is much more detailed than it actually is. Figure 5 shows an implementation of a method called ROAM (Duchaineau et al., 1997), which has been adapted to incorporate foveation. The focus of the viewer's attention is shown in red, which causes the mesh to be retained at a higher level of detail close to that point - see (Watson, 2000) for further details.

An important issue is whether the performance gains won by using a tracking device offset the expense and inconvenience involved. Currently, systems with high spatial and temporal resolution are quite intrusive and expensive, while those that are less expensive and/or intrusive simply cannot produce results at the rate and accuracy required for real-time graphics. However, if the benefits of this technology become well-established and techniques which exploit it are developed, better non-intrusive and low-cost solutions are sure to follow. Alternatively, systems could use models of visual attention to predict where the eye is likely to be directed while watching an animation, and simply use eye-tracking to verify that the model is correct. In (Yee et al., 2001), a saliency map is used to predict the location of the eye while viewing dynamic scenes, and they claim that this compensates for the lack of an eye-tracker. However, they found that when they implemented parafoveal degradation, although significant speedups were achieved when producing the animation, this factor was only useful for the first few viewings of the animation, as people then tended to look away from the predicted regions to explore the field of view more fully. Therefore, any model of this type, even if top-down processes are taken into account, is unlikely to be robust enough for highly interactive situations, such as a game where the player is exploring a virtual environment fully in order to carry out some task.

2.3 Simulation

When objects are being animated, their motions need to be updated by some means for each time-step of the animation. In traditional animation, artists drew objects and characters in a number of key positions, known as key-frames, and human animators filled in the individual images to produce the effect of these objects moving, called in-betweening. In computer-based animation, similar techniques can be used, in that scripts and key-frames are provided for the objects in a scene and the computer simply takes over the role of in-betweening. Such key-framed animations are often used in interactive graphical systems also, for example to add cheap animation effects to characters in a game. However, the fixed nature of such scripted motions restricts interactivity, so other techniques which actually generate the motions automatically are often desirable.

The process of updating the positions and orientations of objects based on the laws of physics is referred to as physically-based animation or simulation. The process of evaluating the current physical state of the objects in the scene and their interactions with each other, and thus determining the appropriate physical response, provides a further significant burden on the graphics engine. In fact, this may be the most significant overhead for many systems, as modern systems relieve much of the rendering burden of by delegating many tasks to specialized graphics acceleration hardware. Because of the time constraints imposed by systems that need interactive frame rates, physically accurate movements are not always possible to generate in the time allowed. Therefore, as in the case of rendering, simulation accuracy must sometimes be sacrificed for speed.

Several researchers have proposed strategies for simplifying simulations while maintaining some degree of physical accuracy. In (Chenney and Forsyth, 1997), objects outside the current view are no longer updated based on physical laws. In (Carlson and Hodgins, 1997), the motions of legged creatures not in view are updated based on simplified rules, while (Barzel et al., 1996; Chenney and Forsyth, 2000) maintain that plausibility as opposed to accuracy is acceptable in many situations, and examine ways in which physically plausible simulations may be generated. However, perceptual principles have to date been largely neglected when considering the issue of level of detail (LOD) for simulation.

One of the most important behaviours of objects that are simulated in a physically accurate or plausible way, is the way that they react when they touch or collide with each other. Without a mechanism to handle such collisions, objects would simply float through each other. Collision handling is, unfortunately, extremely expensive in terms of computational power, and can often be the major bottleneck in physically-based animations consisting of many interacting objects. The avalanche simulation described in (Mirtich, 2000) is an example of such a situation where the collision processing slows down the simulation to a totally non-interactive rate - in this case 97 seconds on average to compute the simulation for just one frame of the animation. In previous studies, we have investigated techniques for graceful degradation of collision handling and the perceptual impact of such degradation, in particular with respect to eccentricity, see (O'Sullivan and Dingliana, 2001; Dingliana and O'Sullivan, 2000; O'Sullivan et al., 1999). To our knowledge, these are the only studies to date that explore the perceptual aspects of level of detail for physically-based simulation in interactive graphics. The issues of collision perception and gaze-contingent collision handling will be explored in more detail in Section 3.

3 Collisions and Eye-movements

While a viewer is watching a simulated collision between two objects, several factors will determine whether they perceive that collision to be "correct", i.e. consistent with their expectations of how those objects should behave. In (O'Sullivan and Dingliana, 2001), we described studies that investigated some of these factors, in particular eccentricity, separation, distractors, causality and accuracy of physical response. These are presented in section 3.1. Eye-movement data could be extremely useful in determining which collisions to resolve at a higher LOD and in Section 3.2 we will describe a new system for gaze-contingent collision handling using an eye-tracker and the results from an initial evaluation experiment will also be presented in Section 3.3.

3.1 Perception of Collisions

Newtonian mechanics can be used to describe the behaviour of objects in the physical world, using dynamic concepts such as force and mass. However, most people have intuitive preconceptions concerning mechanical events that, although incorrect according to Newtonian mechanics, are highly stable and widespread (Clement, 1982). It has also been shown that people use only one dimension of information when making dynamical judgements (Profitt and Gilden, 1989). Therefore, when a dynamic event involves more than one dimension of information, such as velocity and rotation (i.e. an extended body motion as opposed to a particle that has only one dimension of information), humans are less able to correctly identify anomalous physical behaviour. The same authors also discovered that judgements about collisions were made based on heuristics, and that people were influenced by kinematic data, such as velocity after impact and the way that the colliding objects ricochet (Gilden and Profitt, 1989). We maintain that we can exploit this inaccuracy in human perception to produce more visually plausible physical simulations. We wish to determine the circumstances under which this degradation in accuracy will be imperceptible. Robust factors that can significantly affect a viewer's perception of a collision may then be used to prioritise collision processing in a perceptually-adaptive system. Some experiments that investigated these issues are described in (O'Sullivan and Dingliana, 2001).

Causality refers to the ability to detect whether one event causes another. For example, a collision of a moving object with a stationary one will cause the second object to move, whereas a stationary object that starts to move by itself is perceived to be autonomous (Michotte, 1963; Scholl and Tremoulet, 2000). We ran an experiment similar to Michotte's famous causality tests and found that adding a time delay between object contact and collision response reduced the perception of causality and thereby the plausibility of the collision event itself. Therefore, we can conclude that constant frame rates are imperative in any real-time collision handling system and hence interruptible collision detection is the only feasible solution for large numbers of complex objects.

Interrupting collision detection before it is complete either leads to interpenetrations, which are usually unacceptable, or more frequently to objects which bounce off each other at a distance. We found that the separation of objects when they collide provides a strong visual impression of an erroneous collision, but that this effect may be ameliorated by factors such as occlusion of the collision points, eccentricity (i.e. peripheral events) and the presence, number and type of distractors (e.g. visually similar distractors have a stronger masking effect).

We also found that, despite reduced collision detection resolution, it is possible to produce a random collision response that is as believable as the more accurate ones, thus further masking collision anomalies. As in (Profitt and Gilden, 1989), we conclude that people seem to be capable of correctly perceiving errors in collision response only when there is one salient feature (such as gap size), whereas when the simulation becomes more complex, they rely on their own naïve or common-sense judgements of dynamics, which are more often than not inaccurate.

In the following sections, we will discuss strategies for building these factors into a gaze-contingent collision handling system. Further experiments involving the measurement of eye-movements are being conducted to identify the effect of these factors in more complex scenarios with larger numbers of colliding entities and we also present some initial results.

3.2 Gaze-contingent Collision Handling

Collision Handling is an important part of dynamic simulation, as contacts and collisions are the primary source for interaction within the simulation world. Collision handling incorporates three computationally expensive processes which are key candidates for optimisation using an adaptive level of detail approach. The first stage of collision handling involves *detecting* when virtual objects in a simulation scene are in contact or indeed interpenetrating with one another. Should such a condition be found, it is the role of the *contact modelling* process to identify the points or areas of contact. Then the *collision response* module has the twofold duty of resolving any interpenetrations by moving the objects back to a safe position and then applying the correct forces or impulses on the colliding objects to simulate how the objects would behave in the real world.

Adaptive level of detail modulation in collision handling is made possible by using multi-resolution volume models to represent objects in the simulation world. Aside from the display models used in rendering, each object is associated with a functional model to represent its volume. This volume data can then be used as a parameter in all parts of the collision handling phase of simulation. Numerous approaches exist for modelling systems based on multiresolution volume models - see (Hubbard, 1996; Gottschalk et al., 1996; Bergen, 1997) and others. Hierarchies built with spheres are particularly useful for collision detection (Hubbard, 1995). Figure 6 shows an example of a sphere tree representation of an object at different levels of detail.

An interruptible collision handler works by progressive refinement of collision data. Potentially colliding objects are first dealt with at the lowest level of detail

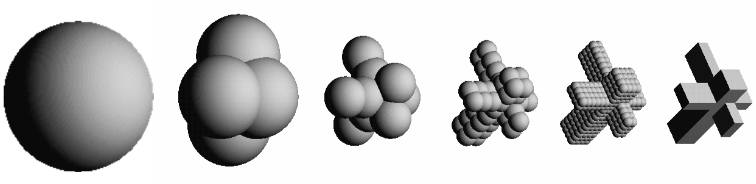


Figure 6: Multiple levels of resolution of a sphere tree model of an object and the original object on the far right.

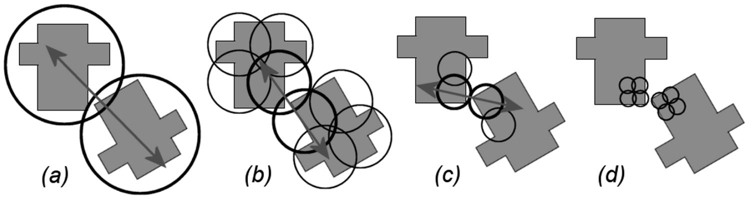


Figure 7: Collision handling data gathered at four different resolutions.

e.g. the bounding sphere (Figure 7(a)). At this level alone the collision handler is able to make a coarse judgement of whether or not two objects are colliding and how they should react to the collision. The circles show the volume model currently processed while the arrow shows the direction of forces calculated at this level. Once this coarse judgement is made and preliminary output is generated, consisting of the new states of the objects (e.g. new velocities and positions), the system has the option of refining the data by inspecting the objects at the next higher resolutions of volume data (figure 7(b), (c), (d)). We can see in figure (d) that in the particular case shown, there is in fact no collision. It should be noted that each higher level of volume data requires progressively more computation time.

A *scheduler* keeps track of the processing time spent on computing the current simulation frame and interrupts this refinement process when the allocated time for collision handling has been exceeded. The simulation process proceeds by using the highest resolution data that the collision handling system was able to compute in the allocated time.

This time-critical approach alone can be used to guarantee real-time frame rates but can be strategically optimised by refining objects to higher levels in regions of the scene that are more important. Different strategies can be implemented for determining the importance of scene regions, for example by taking account of the factors describe in Section 3.1. In a gaze-contingent prioritisation scheme we can use an eye-tracker to determine the user's gaze location at any instant during the simulation. Eccentricity can then be used as the primary measure of importance, or combined with other factors as part of a more complete perceptual model.

Given any prioritisation strategy, we would ideally wish to sort all objects in the scene based on their priority and apply simulation refinement to objects in order of their priority values. However in practice, the computational cost of performing a complete sort can become unjustifiably prohibitive so a more practical approach is to use a small number of priority groups into which colliding objects are bin sorted. Each priority group is then allocated its share of processing time by the scheduler, with more processing being spent on higher priority groups. This method, whilst still preserving some level of prioritisation, bears considerably less overhead expense than a full continuous sort and in practice delivers improvements even with only two priority groups.

3.3 Evaluation

To evaluate the effectiveness of a gaze-dependent prioritisation scheme for interactive simulation, an experiment was performed. Ten participants (computer science staff and students) were presented with 36 short simulations of rigid bodies colliding and bouncing off each other inside a closed cube. The simulation was run on a desktop PC with graphics acceleration, with a 22-inch screen. Participants were instructed to react to the quality of the simulations in two different ways. The first task was to respond, by clicking the mouse button, whenever they perceived the occurrence of a frame containing one or more "bad" collisions during the course of the simulation. A bad collision here refers to one resulting from a coarse level approximation of a collision as described in the previous section. In an initial training phase, they were shown examples of what both good and bad collisions should look like. The second task was to rate the overall quality on a scale of one to five, at the end of each simulation. During the training phase, examples of the best and worst quality simulations were shown, and they were told that the two extremes should receive a rating of five and one respectively. They were also told that simulations with quality ranging between both of these limits were also possible. They then practised on a further number of simulations and were observed to ensure that they had understood the instructions.

Four distinct types of simulations were presented to the participants in random order. The first type of simulation (denoted as *all good*) resolved all collisions at the highest resolution of the volume model. It was possible to deal with objects at this high a resolution in the experiment as the maximum number of objects dealt with was relatively small. A second type of simulation (*all bad*) dealt with all collisions at the very lowest level of resolution i.e. object collisions were dealt with at the bounding sphere level, resulting in objects repulsing each other at a distance in almost all cases. It should be noted that this distant repulsion is not always obvious to viewers as inter-object occlusion sometimes prevents the gap from being visible in the projected display.

Two further types of collisions had combinations of good and bad collisions occurring in the scene at the same time throughout the simulation. In both of these, a *high-priority region* was chosen in the scene where collisions were dealt with at the *all-good* level while outside of this region all objects were dealt with at the coarse level. In one of these, the *tracked* simulations, the users

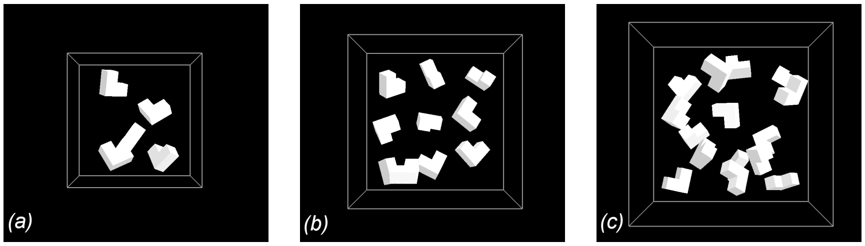
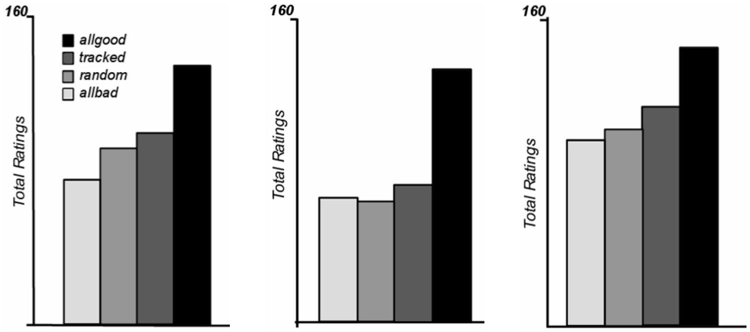


Figure 8: Screenshots of the experiment with 5, 10 and 15 objects contained in proportionately-sized boxes.



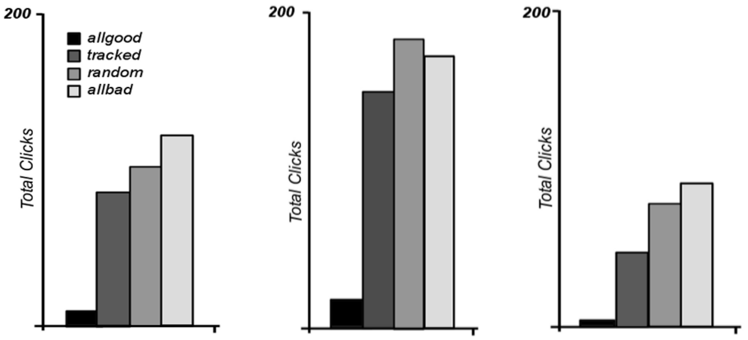
(a) *Five objects* (b) *Ten objects* (c) *Fifteen objects*

Figure 9: Results from rating task

gaze position was tracked and used as the centre of the high-priority region. In the other case, the *random* simulations, a random position was chosen every 5 frames to serve as the centre of the high priority region. Having a randomly located priority region of the same size in the scene ensures that roughly the same proportion of good and bad collisions is maintained as in the *tracked* case.

Each simulation type was shown with 5, 10 and 15 objects and three repetitions each were shown for these twelve cases, see Figure 8. In varying the number of simulated objects, the size of the cube, within which the objects were contained, was correspondingly resized to maintain a constant density of objects at all times within the container. This was in order to ensure consistency in the number of collisions occurring in the simulation. The size of the boxes displayed were not scaled to fill the screen, as then the size of the objects would vary between conditions. Necessarily, this reduced the active field of view for the smaller number of objects.

After a short training phase, in which participants were shown isolated cases of good and bad collisions, participants eye-movements were recorded at all times with an SMI EyeLink eye-tracker and the 36 simulation runs were shown in random order.



(a) Five objects *(b) Ten objects* *(c) Fifteen objects*

Figure 10: Results from clicks task

3.3.1 Results

Figure 9 shows the participants' ratings for the different sets of simulations organised by number of objects in the simulation. Of most interest here, in the context of a gaze-contingent system, is the comparison of the *tracked* and *random* graphs. The results clearly show an overall improvement in the perception of the *tracked* simulation. A single-factor ANOVA showed a weak significance of $> 70\%$, $> 55\%$ and $> 80\%$ respectively for the 5, 10 and 15 object results.

An examination of the number of clicks for each of the simulations shows similar results. There is an overall improvement in the number of clicks in each of the simulation cases with weak statistical significance of $> 60\%$, $> 75\%$ and $> 80\%$ respectively for the 5, 10 and 15 cases. The graphs of the total clicks during the simulations, shown in Figure 10, show a consistent reduction in the number of clicks for the 15 object simulations. It is reasonable to assume that this is due to an increase in the number occluded objects as well as in the number of similar distracters as discussed in Section 3.1.

Why there isn't a stronger statistical significance has to do with the difficulty in setting up fair experimental simulations. The complexity and multi-dimensionality of the simulation process makes it difficult to design an experimental task that is both unambiguous and fairly representative of the variables being evaluated. This is particularly so in the case of gaze-contingent simulation where a random variable (i.e. the gaze position) is an active factor which affects the outcome of the simulation that is given to the participants for evaluation. Modulating simulation detail levels at random or gaze-dependent locations in the scene introduces a significant level of non-determinism into the simulation. This makes it impossible to show all the participants an identical set of simulations. A time of ten seconds was chosen as the duration for each simulation to give the participants a representative sample of collisions.

Some participants reported that they used peripheral vision in certain cases to decide on their rating for the simulation. As discussed in the previous section, a potentially more effective solution would be to have simulation levels degrade progressively with the projected distance from the gaze position. The experiment, however, used a simpler, two-level scheme for prioritisation (i.e. fine resolution within the high-priority region and coarse resolution everywhere else). While there are some studies that examine the ideal size of high-resolution regions for scene viewing, as in (Loschky and McConkie, 2000), there is no documented study that suggests an ideal radius for a high priority region for simulation purposes. Future work is planned to examine the effects of modulating the radius of the high priority region in different simulation cases and the full field of view will also be used, thus allowing more extensive exploitation of the eccentricity effect.

References

Andrews, T. and Coppola, D. (1999). Idiosyncratic characteristics of saccadic eye movements when viewing different visual environments. *Vision Research*, 39:2947–2953.

Aubert, H. and Foerster, O. (1857). Beiträge zur kentniss des indirekten sehens (i): Untersuchungen über den raumsinn der retina. *Arch. Ophthamology*, 3:1–37.

Barzel, R., Hughes, J., and Wood, D. (1996). Plausible motion simulation for computer graphics animation. In *Computer Animation and Simulation'96*, pages 183–197. Springer-Wien.

Bergen, G. V. D. (1997). Efficient collision detection of complex deformable models using aabb trees. *Journal of Graphics Tools*, 2(4):1–13.

Carlson, D. and Hodgins, J. (1997). Simulation levels of detail for real-time animation. In *Proceedings Graphics Interface*, pages 1–8.

Carrasco, M. and Frieder, K. (1997). Cortical magnification neutralizes the eccentricity effect in visual search. *Vision Research*, 37(1):63–82.

Chenney, S. and Forsyth, D. (1997). View-dependent culling of dynamic systems in virtual environments. In *ACM Symposium on Interactive 3D Graphics*, pages 55–58.

Chenney, S. and Forsyth, D. (2000). Sampling plausible solutions to multi-body constraint problems. In *Proceedings Siggraph 2000*, pages 219–228.

Clement, J. (1982). Students' preconceptions in introductory mechanics. *American Journal of Physics*, 50(1):66–71.

DeValois, R. and DeValois, K. (1988). *Spatial Vision.* Oxford University, New York.

Dingliana, J. and O'Sullivan, C. (2000). Graceful degradation of collision handling in physically based animation. *Computer Graphics Forum (Eurographics 2000 Proceedings)*, 19(3):239–247.

Duchaineau, M., Wolinsky, M., Sigeti, D., Miller, M., Aldrich, C., and Mineev-Weinstein, M. (1997). Roaming terrain: Real-time optimally adapting meshes. In *Proceedings of IEEE Visualization '97*, pages 81–88.

Funkhouser, T. and Séquin, C. (1993). Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *Proceedings SIGGRAPH '93*, pages 247–254.

Garland, M. (1999). *Quadric-Based Polygonal Surface Simplification.* PhD thesis, Carnegie Mellon University, Tech. Rept. CMU-CS-99-105.

Gilden, D. and Profitt, D. (1989). Understanding collision dynamics. *Journal of Experimental Psychology: Human Perception and Performance.*, 15(2):372–383.

Gottschalk, S., Lin, M., and Manocha, D. (1996). Obb-tree: A hierarchical structure for rapid interference detection. In *Proceedings of ACM SIGGRAPH '96*, pages 171–180.

Grimes, J. (1996). On the failure to detect changes in scenes across saccades. In Akins, K., editor, *Perception (Vancouver Studies in Cognitive Science)*, volume 2, pages 89–110. Oxford University Press.

Henderson, J. and Hollingworth, A. (1999). The role of fixation position in detecting scene changes across saccades. *Psychological Science*, 5:438–443.

Hubbard, P. (1995). Collision detection for interactive graphics applications. *IEEE Transactions on Visualization and Computer Graphics.*, 1(3):218–230.

Hubbard, P. (1996). Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210.

Janott, M. (1999). *Interactive Perception of Multiresolution Meshes.* Final year thesis, Computer Science Dept. Trinity College Dublin.

Janott, M. and O'Sullivan, C. (1999). Using interactive perception to optimise the visual quality of approximated objects in computer graphics. In *10th European Conference on Eye Movements, Book of Abstracts*, pages 110–111.

Loschky, L. and McConkie, G. (2000). User performance with gaze contingent multiresolutional displays. In *Proceedings of the Eye Tracking Research and Applications Symposium*, pages 97–103.

Luebke, D., Hallen, B., Newfield, D., and Watson, B. (2000). Perceptually driven simplification using gaze-directed rendering. Technical report, CS-2000-04 University of Virginia.

Michotte, A. (1963). *The Perception of Causality.* Basic Books., New York.

Mirtich, B. (2000). Timewarp rigid body simulation. In *Proceedings SIGGRAPH 2000*, pages 193–200.

Murphy, H. and Duchowski, A. (2001). Gaze-contingent level of detail rendering. In *Proceedings of EuroGraphics 2001 (Short Presentations)*, pages 219–228.

Ohshima, Yamamoto, and Tamura (1996). Gaze-directed adaptive rendering for interacting with virtual space. In *Proceedings IEEE VRAIS'96.*

O'Regan, J. (2001). Thoughts on change blindness. In Harris, L. and Jenkin, M., editors, *Vision and Attention.*, pages 281–302. Springer.

O'Sullivan, C. and Dingliana, J. (2001). Collisions and perception. *ACM Transaction on Graphics*, 20(3):–.

O'Sullivan, C., Radach, R., and Collins, S. (1999). A model of collision perception for real-time animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *Computer Animation and Simulation'99*, pages 67–76. Springer-Wien.

Profitt, D. and Gilden, D. (1989). Understanding natural dynamics. *Journal of Experimental Psychology: Human Perception and Performance.*, 15(2):384–393.

Rayner, K. (1986). Eye movements and the perceptual span in beginning and skilled readers. *Journal of Experimental Child Psychology*, 41:211–236.

Reddy, M. (1997). *Perceptually Modulated Level of Detail for Virtual Environments.* PhD thesis, University of Edinburgh.

Rensink, R., O'Regan, J., and Clark, J. (2000). On the failure to detect changes in scenes across brief interruptions. *Visual Cognition*, 7(1):127–146.

Rovamo, J. and Virsu, V. (1979). An estimation and application of the human cortical magnification factor. *Experimental Brain Research*, 37:495–510.

Scholl, B. and Tremoulet, P. (2000). Perceptual causality and animacy. *Trends in Cognitive Sciences.*, 4(8):299–309.

Shioiri, S. and Ikeda, M. (1989). Useful resolution for picture perception as a function of eccentricity. *Perception*, 18:347–361.

Simons, D. (2000). Current approaches to change blindness. *Visual Cognition*, 7(1/2/3):1–15.

Strasburger, H., Rentschler, I., and Jr., L. H. (1994). Cortical magnification theory fails to predict visual recognition. *European Journal of Neuroscience*, 6:1583–1588.

Triesch, J., Sullivan, B., Hayhoe, M., and Ballard, D. (2002). Saccade contingent updating in virtual reality. In *ETRA'02, Symposium on Eye-tracking Research and Applications*, page To Appear.

van Diepen, P., Wampers, M., and d'Ydewalle, G. (1998). Functional division of the visual field: Moving masks and moving windows. In Underwood, G., editor, *Eye guidance in reading and scene perception*, pages 337–355. Oxford: Elsevier.

Watson, M. (2000). *Applications of Gaze Tracking*. Final year thesis, Computer Science Dept. Trinity College Dublin.

Weymouth, R. (1958). Visual sensory units and the minimal angle of resolution. *American Journal of Ophthamology*, 46:102–113.

Yee, H., Pattanaik, S., and Greenberg, D. (2001). Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Transactions on Graphics*, 20(2):39–65.