# A Model of Collision Perception for Real-Time Animation

*Carol O'Sullivan*
*Image Synthesis Group, Computer Science Department, Trinity College Dublin*
*Email: Carol.OSullivan@cs.tcd.ie*

## Abstract

A model of human visual perception of collisions is presented, based on two-dimensional measures of **eccentricity** and **separation.** The model is validated by performing psychophysical experiments. We demonstrate the feasibility of using this model as the basis for perceptual scheduling of interruptible collision detection in a real-time animation of large numbers of homogeneous objects. An **eye-tracker** is used to locate the user's point of fixation. By using a priority queue scheduling algorithm, perceived collision inaccuracy was approximately halved. The ideas presented are applicable to other tasks where the processing of fine detail leads to a computational bottleneck.

## 1 Introduction

The aim of interactive animation systems is to create an exciting and real experience for viewers, to give them a feeling of immersion, of “being there". The tendency in the past has been to attempt to achieve this by matching as closely as possible the physics of the real world, with varying degrees of success. Of course, what a person perceives is strongly affected by the physical behavior of the world around them, but it is the human visual system that receives and interprets the visual cues from the surrounding environment, and it ultimately determines what we perceive. Therefore, we must look beyond the laws of physics to find the secret of reproducing visual reality.

In interactive animation applications such as VR or games, it cannot be predicted in advance how a user or the entities in a virtual world will behave, so the animation must be created in real-time. As the number of independently moving objects in the scene increases, the computational load also increases. Possible scenarios are crowd simulations or rockfalls, where large numbers of entities move around a virtual world in real-time. There are many bottlenecks in such systems, collision detection being a major one. A trade-off between detection accuracy and speed is necessary to achieve a high and constant frame-rate. However, it is possible to reduce perceived inaccuracy by taking perceptual factors into account, and also by using an **eye-tracking device** to locate where on the screen a viewer is looking.

A model of human visual perception of collisions is developed, based on two-dimensional measures of **eccentricity** and **separation**. We demonstrate how such a model can be used in a real-time, adaptive collision detection algorithm to reduce the perception of collision-handling inaccuracies when animating large numbers of objects. The model is validated by performing psychophysical experiments. The ideas presented are applicable to other tasks where the processing of fine detail leads to a computational bottleneck.

### 1.1 Background

Traditional collision detection algorithms have required a large amount of geometrical intersection tests. To improve the efficiency of such algorithms, hierarchical representations of entities were developed to localise the areas where the actual collision occurred. These include sphere-trees [14][20][21], OBB-trees (Oriented bounding boxes) [11], and hierarchies of *k*-DOPs (Discrete Orientation Polytopes) [17]. While speed and efficiency has been the main focus of such research, the issue of a constant frame rate is also paramount. This problem has been addressed in part by exploiting coherence [5], and by using an interruptible collision detection algorithm [14]. The advantage of an interruptible algorithm is that the application has full control over the length of time that the collision detection algorithm may take. It can then use this to control frame rate, keeping it constant and high. The disadvantages of this approach are that inaccuracies in collision detection may cause the viewer to perceive unrealistic behaviour of colliding entities. We also propose using such an interruptible algorithm, but we will schedule collision processing based on a model of human collision perception.

***Figure 1. The eye-tracker***

### 1.2 Visual Perception and Eye-Tracking

In recent years the realisation has been growing within the computer graphics community of the advantages to be gained by using knowledge of human perception. Perceptual factors such as size and speed of objects have been used to choose the levels of detail (LOD) at which to render objects in a scene [10]. The advantages of simulating plausible motion, as opposed to physically accurate motion, have been investigated [3]. The results of psychophysical research have been used to develop algorithms and models of visual perception for realistic image synthesis [8][9][13]. We maintain that an extension of this approach to a study of human visual perception of dynamic scenes would be very beneficial in solving some of the problems of real-time animation systems. In particular, the analysis of human visual perception of collision events could enable a prioritisation of potential collisions to process within a given frame of an animation, hence reducing the negative impact of interruption.

It has long been established that many visual processing tasks deteriorate at increasing eccentricities from the fixation point [2] [27]. Therefore, an eye-tracking device that locates where on the screen a viewer is looking could be an important tool in such real-time systems: *(eye-tracking) is a promising long term solution, since gaze direction can be exploited for other purposes such as identifying the region of screen space - corresponding to the foveal portion of the retina - that deserves to be rendered with high spatial detail* [6].

In the past, the most common use of eye-trackers has been in medical and scientific research. These types of trackers are very accurate, but also very invasive, involving the use of head restraints, bite bars, or scleral coils which are inserted directly into the eye. More recently, more mobile, non-intrusive, and low-cost trackers have been developed, and their use has been gaining increasing support in the fields of Human Computer Interaction (HCI) and Virtual Reality (VR) [16][24]. We use a small head-mounted tracker from Vision Control Systems, which acts exactly like a mouse (see Figure 1).

The following scenarios are possible when considering the problem of collision detection:

- Do completely accurate collision detection. This will give frame-rate problems.
- Do interruptible collision detection (as in [14]), with no perceptual sorting: This produces a good frame-rate, but bad perception.
- Add an eye-tracker, and use a perceptual model to schedule collision processing. Validate and refine the model with data from psychophysical experiments. This should yield the best possible perception of collision events, and a good frame-rate.

In this paper, we take the latter approach. In Section 2, we describe the 3-D application that has been developed to test the concepts presented. In Section 3, a model of collision perception is presented, and Section 4 analyzes the performance of the application when the model is used for collision scheduling. Section 5 discusses the psychophysics and neurophysiology of collision perception, and Section 6 describes the psychophysical experiments carried out to validate our model. Section 7 presents conclusions and plans for future research.

## 2 The Application

To test the feasibility of the ideas presented herein, we have implemented a three-dimensional animation system where non-convex, star-shaped entities move around and interact in real-time within a volume. The starting position, translational and rotational velocity for each entity is randomized. A simple volumetric representation has been used for the entities, but the collision detection routines are designed to work with any sphere-trees generated from any type of model. Because the focus of this paper is collision detection, collision response is also quite simple, with the velocities of colliding entities simply being exchanged.

We use the “Sweep and Prune” algorithm from [5] as the broad-phase of our algorithm to detect overlaps of the fixed-size bounding boxes of objects. When the bounding boxes of two objects overlap, a **collision** object is created. For the narrow phase of our algorithm, we have developed an interruptible algorithm based on sphere trees, adapted from [20] and [14]. We approximate each entity with a sphere tree during a pre-processing phase. The sphere trees are built using an octree generation algorithm, and we generate 4 levels of sphere trees for each entity in a pre-processing phase (see Figure 2). Initially, the sphere-trees are centered at the origin. Whenever we wish to test for the intersection between two entities, we apply the appropriate transformation to the centres of each pair of spheres we wish to test as required. Each collision object contains information on its current state i.e. what spheres on one tree must be tested against what spheres on the other tree. The algorithm is fully interruptible, allowing the detection to descend one level of one tree at a time, reducing the complexity of the algorithm, and enabling a fast, albeit approximate, response when necessary. There are four possible results at each iteration of the test for a collision object:

1. An intersection is detected between two leaf spheres of the trees. In this case the objects are deemed to be colliding, and the collision is resolved.
2. No intersections are detected between any spheres. In this case, the objects are definitely not colliding, and the collision is resolved.
3. An intersection is detected between a pair of spheres, but at most one is a leaf, so the collision remains active for further processing.
4. During the iteration, the application indicates that it wishes to interrupt collision processing. In this case, if there are remaining spheres to be tested, the entities are deemed to be colliding, and the collision is resolved.

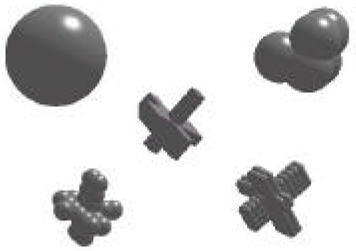


***Figure 2. An entity, and 4 levels of its sphere tree***

Two or more linked lists of collisions are maintained:

- The **active collision list(s)** of pairs of entities suspected of colliding due to an overlap of their bounding boxes, which need further processing to determine whether they are really colliding or not.
- The **real collision list** of pairs of entities deemed to be really colliding during the narrow phase.

The main execution loop is as follows:

```
DO
-   Update the position of all objects
-   Broad-phase: test for overlapping bounding boxes and
    generate the active collision lists
-   Narrow phase: Process the active collision lists,
-   Process the real collision list: compute appropriate
    collision response for each colliding pair of objects.
-   Render Frame
UNTIL animation is terminated.
```

During the narrow phase, as collisions on the active collision lists are resolved, they are placed on the real collision list if a real collision was detected, or are destroyed if it has been determined that the entities are definitely not colliding. At any point in time during collision processing, there will be unresolved collisions on the active list, and resolved collisions on the real list. At some point the application will deem that collision processing should stop, e.g. when a pre-defined target time has been exceeded. Any collisions still being processed will be deemed colliding, and this will leave us with a list of real collisions and a list of unresolved collisions still on the active collision list. We have chosen to treat these collisions as real collisions, so all active collisions are then added to the end of the real collision list.

The key to controlling the collision inaccuracy perceived by a viewer in a given frame of an animation lies in the scheduling method adopted. In [14] active collisions are resolved in round-robin order, descending one level in the hierarchy of every sphere tree at each iteration of the algorithm, until interruption. However, no account is taken of the perceptual importance of each collision. We will call this strategy ***round-robin scheduling***. Another strategy, ***sequential scheduling***, is to start at the first collision and fully resolve each collision in turn, until completion or interruption, but again perception is ignored. In ***perceptually sorted sequential scheduling*** a perceptual importance is attached to each collision, and the active collision list is sorted based on this priority using a version of quick-sort adapted for linked lists. Now when the list is sorted sequentially, collisions which are most important perceptually will be resolved first, leaving the more unimportant collisions to be resolved only if there is time left. Finally, we can generate not one active collision list, but a set of priority queues, and round robin within them. A higher priority queue is resolved first, and only when all collisions on that queue have been resolved is the next highest queue processed. This is called ***priority queue scheduling.***

Our application has been designed to support any of these scheduling strategies. For the perceptually sorted sequential scheduling, it simply requires a function that returns a priority for a given collision, and it will sort all collisions based on this value. In order to set this priority, however, we need to understand how humans perceive whether two objects have collided or not. We also need metrics to determine collision inaccuracy in each frame in order to evaluate the effectiveness of our techniques.

## 3 A Model of Collision Perception

The application provides a framework within which different collision detection, prioritisation and scheduling algorithms may be implemented and evaluated for computational and perceptual performance. We now develop an initial model of collision perception, to be used both to prioritise collisions, and also to estimate perceived inaccuracy. In this way, we can test the feasibility of our approach, and also focus on the type of psychophysical data we wish to gather. When considering the inaccuracy present in a frame of an animation, we must distinguish between **geometrical inaccuracy ∇**, and **perceived inaccuracy P**. The geometrical inaccuracy in a scene is an estimate of the overall three-dimensional error that has been incurred by accepting non-collisions as real, causing entities to repulse without touching. If N is the number of collisions, and $g_i$ is the maximum gap in collision i, we may estimate this error by regarding it as a function of the potential gaps left during such "non-collisions":

$$\Delta = \sum_{i=1}^{N} g_i$$

In our applications, we cannot calculate the exact size of the biggest gap between two entities, as this would take an excessive amount of time and defeat the purpose of approximate collision testing. Instead, we can use the information available to us to estimate an upper bound on the maximum gap size between two entities. We use the three-dimensional distance between the centers of the last two spheres found to be intersecting from the sphere-trees of each colliding pair, or the distance between the centers of the two entities if collision testing is interrupted before any spheres have been tested. Hence, the further down the sphere-tree hierarchy each collision is allowed to progress, the more accurate the estimate will become. Alternatively, we could pre-compute the Hausdorff distance for all spheres in each tree (as in [14]), and use the sum of these distances as our estimate.

Not all collision inaccuracies contribute equally to the inaccuracy perceived by the user in a single frame of an animation. Hence, the perceived inaccuracy **P** present in two frames of an animation with identical geometrical inaccuracy **∇,** may be quite different depending on how the frame is viewed. For now, let us assume that eccentricity **e**, i.e. distance from the viewer's fixation point, and separation, estimated by maximum gap size **g**, are the only two factors which affect perceived inaccuracy. If two spheres are interpenetrating, we find the midpoint on the line segment inside the intersection. We call this the ***Centre of Collision***. We track the user's gaze, so we know the fixation point ***f*** for each frame, expressed as an x,y location on screen. We can therefore calculate the eccentricity **e** as follows: We find the x,y location in screen co-ordinates of the center of collision projected onto the view-plane, then e is simply the 2-dimensional distance from ***f*** of the center of collision. Collisions closer to the fixation point contribute more to the perceived inaccuracy of a frame than those further away and hence should receive higher weighting. Collisions closer to ***f*** should receive lower weighting.

Similarly, the size of the maximum on-screen gap, **g**, may also be used to weight each collision, with larger gaps contributing more to inaccuracy than smaller ones. We calculate an upper bound on the 2-dimensional gap size as follows: We take the centers of the last two spheres found to be intersecting, and calculate the 2-dimensional distance between their projections onto the screen. If N is the number of collisions in a scene, $g_i$ is the 2-dimensional maximum gap size between the entities in collision i, and $e_i$ is the eccentricity of the center of collision i from the fixation point *f*, a possible estimate of **P** is:

$$\Delta = \sum_{i=1}^{N} F(g_i, e_i)$$

The exact relationship between eccentricity and gap-size remains to be determined by psychophysical means, and indeed it is highly probable that these factors alone are not sufficient to estimate perceived collision inaccuracy. However, we have hypothesized what this relationship might be for the purposes of our feasibility tests, and then used the results of these tests to direct psychophysical investigations. For constants C, $C_1$, $C_2$ and $C_3$, possible estimates of the above function **F** are:

$$F_1(g_i, e_i) = \begin{cases} g_i / e_i & \forall \, ei > 0 \\ e_i & \forall \, ei = 0 \end{cases}$$

$$F_2(g_i, e_i) = \frac{g_i}{\exp(e_i / C)} \qquad \forall \, ei$$

$$F_2(g_i, e_i) = \frac{g_i}{\exp(e_i / C)} \quad \begin{array}{ll} C = C_1 & \forall \quad ei < 20 \\ C = C_2 & \forall \quad 20 \le ei < 100 \\ C = C_3 & \forall \quad ei \ge 100 \end{array}$$

These functions provide plausible models of how the visual system might work. They all assume a fall-off in the ability to detect collision anomalies with increasing eccentricity and decreasing gap-size. They differ only in the rate of fall-off, and the relationship between gap-size and eccentricity. Function **F1** decreases very rapidly up to a value corresponding to less than a centimeter on the screen. This assumes a very dramatic fall-off in human ability to detect gaps at increasing eccentricity from the fixation point, and in certain circumstances, this fall-off may be more gradual. In addition, with the use of a highly accurate and intrusive eye-tracking device, we can guarantee that the estimated fixation location is within a millimeter or two of the actual fixation. However, if we use a tracker with lower spatial resolution, and do not employ the use of bite-bars and other such restraints, the best accuracy we can achieve may only be within a circle of several millimeters diameter. Therefore, the above model may be too refined for our purposes. Nevertheless, if the visual and mental task at the viewer's point of fixation is very complex, such a "tunnel vision" effect as described by the perceptual function $\mathbf{F_1}$ may be evident. Function **F2** allows a more gradual fall-off (see Figure 3), and different values of the constant C allow us to control the shape of the function, as does function **F3**, which allows us to "patch together" different functions.

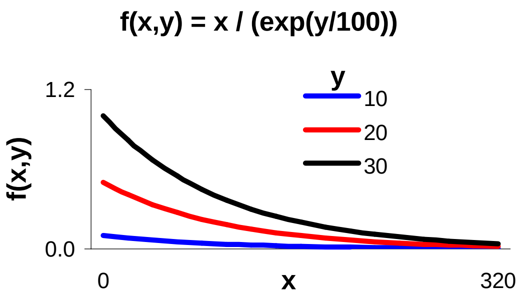


***Figure3. Perceptual function F2***

It is not yet apparent which, if any, of the above models is the most appropriate. In fact, the final model will almost definitely be more complex than the ones we have just proposed. However, it is now possible to test our application with respect to a plausible model of perception, allowing us to manipulate frame times and scheduling mechanisms, and test their effect on the viewer, as measured by our hypothesized perceptual metric. At a later stage, a more refined model of perception of collision inaccuracy based on psychophysical data can be used. As an initial pass at estimating the perceptual importance of a collision in our application, we used the metric **P** just described. It is possible to choose any of the three estimates of F to achieve this. In addition to eccentricity and separation, this metric takes account of three additional factors that could affect collision perception:

- Distance from the viewpoint: P is a 2-dimensional metric, calculated from the perspective projection of 3-d intervals onto a 2-dimensional plane. Thus, the further away a collision is, the smaller the gap will be.
- Size of objects. The larger the objects are, the larger the gap between them will be.
- Visibility of collision points/ angle between the colliding entities. If two entities are oriented towards each other along a line orthogonal to the viewer, it will not be possible to tell whether they are actually touching or not, because one of the objects will occlude the points of collision. In this case, the gap between them when projected onto the view-plane will be zero. If however, the objects are oriented towards each other on a plane parallel with the view-plane, the gap between them will be most noticeable. In this case, the projection of the gap will be at a maximum. As the angle between objects changes from orthogonal to parallel, the projected gap will also continuously increase from 0 to the maximum.

## 4 Performance

To test performance, we ran the application with 10, 30, 100, 300, and 500 objects respectively. Each test was run with no graphics, and only the time taken to perform collision handling was measured. This included broad-phase testing, generation of the active collision list, detection of collisions, and collision response determination. The experiments ran on a Pentium PC, 233 MHz, running Microsoft Windows '95. They were given real-time priority to minimize the disruption caused by other processes taking up CPU time. We used the Win32 system call: GetCurrentTime() to measure the time elapsed in milliseconds.

The object shapes and sizes were identical in each test, as was the density of objects inside each volume. We wanted to create a worst-case scenario, so objects were tightly packed into the volume i.e. 10 objects in a volume of size 1x1x1 ensured a high number of collisions at each frame. Therefore, we increased the size of the volumes for larger numbers of objects proportionally. e.g. 30 objects were packed into a volume of size 1.44, i.e. the cubed root of 3, and so on. The size of the screen viewport was identical in each case. The front of each volume was always the same distance from the viewpoint. We recorded results from 5000 frames for the given number of objects. In addition to recording the time spent on collision handling: **T**, the overall inaccuracy: $\Delta$ , and the perceived inaccuracy: **P**, for each frame, we also recorded the number of broad-phase and real collisions, and the number of narrow phase tests that were resolved at each of levels 1, 2, 3 and 4 of either sphere tree.

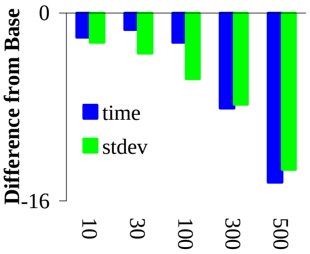
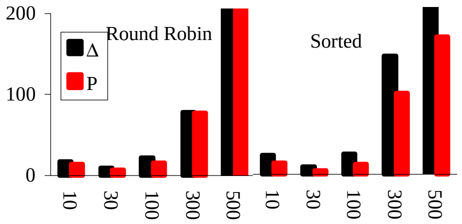


*Figure 4. Reductions in collision handling times with interruptible collision detection.*

We first ran all experiments with no interruption. In order to reach an overall target frame rate of at least 10 frames per second, we estimated that at most 50 milliseconds should be spent on collision handling per frame. In the case of 100, 300 and 500 objects, this was not achieved in the base case. In the 10 and 30 object tests, the frame-rate standard deviation was quite high. In order to estimate the time needed for the non-detection activities of our collision handling functions, we repeated all tests, this time interrupting at 0, i.e. only performing broad-phase tests and collision response. Based on these results, we chose a target time X for each number of objects, and interrupted collision detection at this time.



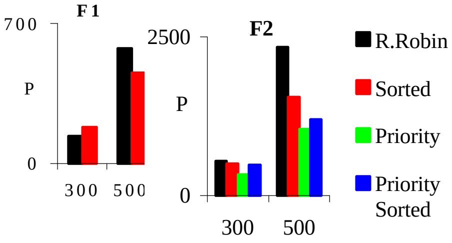
(a) (b)

*Figure 5. Increases in inaccuracies with interruptible collision detection.*

We found that optimal results were achieved by interrupting the 10, 30, 100, 300 and 500 object animations after 5, 20, 44, 30 and 10 milliseconds respectively. Figure 4 shows the reduction in mean time and standard deviation achieved by interrupting collision detection, compared to the base case. This improvement is very significant, up to a factor of -16 in the 500-object case. Figure 5(a) shows the increase in both geometrical and perceived inaccuracy for the same tests. For up to 100 objects, the increase in inaccuracy is not too damaging, but as the number of objects increases, both the perceived and total inaccuracy increases significantly, up to 200 times worse. Some level of inaccuracy must be accepted as a trade-off for shorter and less variable collision-handling times. However, we can attempt to decrease the perceived inaccuracy by using the perceptual metrics described in Section 4 to schedule the further processing of collisions detected in the broad phase.

Next, we ran the same tests using ***perceptually sorted sequential scheduling***, using function $\mathbf{F_1}$ both to measure inaccuracy, and to schedule the processing of collisions. We can see from Figure 5(b) that in all cases, the increase in perceived inaccuracy, **P**, has been decoupled from the increase in geometrical inaccuracy **Δ**. However, the added overhead of performing a quick-sort at each frame has reduced the time available for collision detection, and hence in the 300 object case, has caused an increase in **Δ**, so although the relative increase in **P** is lower, its absolute value is actually higher than in the round-robin case. This can be seen from figure 6 (a), which shows the absolute values of **P** for both scheduling mechanisms. In the 500 object case, there is so little time available for collision detection when interrupting after 10 milliseconds, that the already very high geometrical inaccuracy is not much higher, whereas the perceived inaccuracy is slightly improved.

We speculated in section 2 as to whether function F1 was the most appropriate to model the human visual system. It is also possible the function F2 is more appropriate. To this end, we repeated the above tests for both 300 and 500 objects, but this time using function F2 both to measure inaccuracy and to sort collisions. We can see from figure 6(b) that based on this model, there is an improvement in perceived inaccuracy when sorted scheduling is used, and this is most noticeable in the 500-object case. (Note, the F2 metric has a different scale, and cannot be compared directly with the F1 values). Next, we ran the tests using priority queue scheduling. We achieved this by simply setting an eccentricity from the fixation point, and creating one active list of collisions inside this region, and another of those outside the region. We tried two approaches: priority only, and priority sorted. In the former case, the collisions on the first active list are resolved in simple round-robin fashion, and only when they are all resolved are the collisions on the second list processed. In the second case, the first priority list is sorted and processed sequentially, and the second list is processed in round-robin order, thus reducing the sorting overhead incurred in the fully sequential case. The results are shown in Figure 6(b). It is clear from these results that the simple priority queue scheduling produced the best results for both 300 and 500 objects, approximately halving inaccuracy in both cases.



(a) (b)

*Figure 6. Comparison of different collision scheduling strategies.*

## 5 Neurophysiology and Psychophysics of Collision Perception

Psychophysics is a set of techniques used to study mappings between events in the environment and levels of sensory responses, by conducting non-invasive experiments on humans. Neurophysiology is the search for explanations of these perceptual phenomena by examining actual neural mechanisms, most commonly of animals with cortical architectures similar to humans. Both approaches complement each other: psychophysical results are more secure when they can be linked to known physical structures, and the work of neurophysiologists is often motivated by the need to explain phenomena discovered in psychophysical research.

There are many potential factors that affect a human's ability to notice whether two objects have collided realistically or not. To consider using the factor in a prioritisation algorithm, the effects observed should be significant, and occur in most, if not all, of the subjects tested. In addition, they must be **robust**. There is no point in using a factor that only occurs under some highly specialised conditions. We are looking for factors that can be generalised over a wide range of conditions, because we wish to apply them in a real-world scenario. There is an infinity of variable combinations and possible experiments, so following some exploratory experiments, we have picked a set of conditions that are useful to investigate the principal questions. Our strategy is to follow one line in the space of possibilities, which can later be extended and refined. In the applications being considered, three types of collisions may occur:

- "True" collisions, where entities touch, the collision is detected, and fully accurate collision response occurs. We may consider this as being the control situation for experimental purposes.
- Interpenetrations, where the entities also touch, but the collision is not detected or is ignored by the application. The entities are therefore allowed to continue on their previous path, even though it causes them to merge into each other to a greater or lesser degree.
- Repulsions, where the entities are close to each other but have not actually touched. In this case the application decides to take the chance that they are actually touching, and accepts this situation as a true collision, causing a repulsion effect.

There are certain points to be made in favour of allowing only repulsions to occur. The effect of one entity piercing through another is very noticeable and observations strongly suggest that this effect is more disturbing than the effect of repulsion, especially if the entities are of different colours. Another problem with interpenetration is that the anomaly lasts longer than repulsion, i.e. if two interpenetrating entities are ignored for several frames, they will continue to interpenetrate further and further, hence increasing the chance that they will be observed by a viewer. In addition, the visual perception of repulsion has well documented parallels in the study of spatial vision, hyperacuity, and brain physiology. A summary of the physiological reasons for decreased spatial resolution in the periphery appears in [7]:

- Information projected onto the central part of the retina, i.e. the fovea, receives more processing, because there are more cones concentrated there.
- There is an almost one-to-one correspondence between the photo-receptors in the fovea, and the ganglion cells there. In the periphery, hundreds of photoreceptors can converge onto just one ganglion cell.
- The receptive field sizes of cells in the fovea are smaller than in the periphery i.e. the area of the visible scene which a foveal cell must process is smaller than those cells at greater eccentricities.
- In the visual cortex, there is more representation for the fovea than for the periphery, allowing for more acute visual processing (cortical magnification).

This means that detection of fine detail is facilitated in the fovea, and deteriorates with **eccentricity**. Therefore, it is likely to affect the ability to detect a collision anomaly. **Separation** or gap-size will also most likely be a factor, as there is a one-to-one mapping from the cells in the retina to the cells in the primary visual cortex, called a retinotopic mapping, and it is quite precise, enabling spatial location information to be efficiently processed [25]. However, other factors may also affect collision perception, such as the **location** of the entities in the periphery**.** The eccentricity effect is not always symmetric. It was found that a 75 per cent confidence region for a visual search task was elliptic, with performance being better in the horizontal regions rather than in the vertical [19]. The visual cortex contains many cells that are orientationally selective, i.e. they perform best when the stimuli in their receptive fields are oriented at a particular angle [15]. The **direction of offset** of two stimuli has been found to be a significant factor when performing a detection task in the periphery, and more important that the orientation of the stimulus itself [28].

**Image motion** has been observed to have a degrading effect on various types of hyperacuity task. In [4] they tested the effect of stimulus velocity on Vernier acuity (the ability to detect offsets between two lines), and discovered that performance worsens as the velocity increases. They explain this by a shift of sensitivity to mechanisms of lower spatial frequency. In other words, the faster a stimulus moves, the less fine detail the retina and hence the visual cortex can determine. In addition to orientationally-selective cells in the visual cortex, there are also cells which are sensitive to **direction of motion** in the area of the cortex considered to be primarily responsible for motion detection, i.e. the middle temporal visual area (MT) (also called area V5) [29]. A "centrifugal directional bias" has been found in the brain of the macaque monkey, where there were more cells responsive to directions away from the fovea than any other direction [1]. A suggested reason for this is that as an animal moves, the visible scene is constantly expanding away from the fovea. Cells have also been found in another area of the brain (MST) which are responsive to certain types of spiral motion [12].

The presence or absence of **distractors** is also highly likely to affect a human's ability to accurately detect collision or non-collision, as is the nature of the distractors. These issues have been extensively researched in the area of Visual Search [22][26]. If the distractors are in a clearly distinguishable perceptual grouping from the target to be searched for, the identification of this grouping occurs automatically, without any attention or search being necessary. Such a grouping may be of distractors of similar colour, orientation or common movement, which differ from the target. This is referred to as a **preattentive pop-out task**. If such an obvious grouping is not immediately apparent, it is necessary to focus attention on each item in turn, and this is called a **serial search task**. In such a task, performance is significantly worse than in the pop-out tasks. In the tasks that we are considering, e.g. simulations of large numbers of homogeneous interacting entities, such as crowd scenes or rockfalls, there will not be obvious perceptual groupings of objects. Therefore, the ability of viewers to detect a collision anomaly, i.e. a repulsion, in such a scenario is of major interest to us.

Other important factors which we do not consider here may also affect the perception of collisions, such as size, velocity, acceleration, colour, shape, visibility and semantics. Incorporation of all these factors into our model may make it overly complex, and impact on the performance of the system. However, it is intended to extend this work to evaluate their usefulness.

## 6 Psychophysical Experiments

We carried out 4 sets of psychophysical experiments. 11 students aged between 18 and 21 were the subjects (3 female, 8 male), and all had normal corrected or uncorrected vision. In experiment 1, we tested detection performance with no motion, in 2 we added motion, in 3 we added distractors that were different from the colliding entities i.e. a "pop-out" task, and in 4 we added distractors which were identical to the colliding entities, i.e. a "serial search" task. A 2-dimensional framework for conducting experiments has been developed which is easy to configure to generate the desired collision events. In all experiments some common presentation conditions were maintained. Motion is 2-dimensional, which does not impact upon the generality of our results, since it has been shown that humans use only two-dimensional visual information to make decisions about collision events, such as time to collision [23].

The stimuli are white circles on a black background, ideal because they have no orientation, and the distance between two of them is independent of their direction of offset. The monitor was 22-inch, and the experiments took place in a darkened room, eliminating screen glare. In each experiment, there was a 2-second delay between presentation of each stimulus. A small cross remained in the centre of the screen for the purpose of fixation. Viewing distance was held constant at 70cm, and the left and right mouse buttons were used to indicate collision or repulsion respectively. Subjects were encouraged to take frequent short breaks to avoid eye strain and blurring, and there was a trial run of each experiment, to familiarise the subject with the methods and stimuli.

In Experiment 1, stationary stimuli were presented to the subjects at 3 different eccentricities, 4 different directions of offset, and 8 locations. 50% of the runs were of touching circles, in 25% of cases they were separated by a small gap, and in the remaining 25% of cases, they were separated by a larger gap. Each run was replicated 3 times. The stimuli were presented for 150 milliseconds only, preventing eye-movements. Experiment 2 was identical, except that the stimuli were now in motion, and approached each other, and moved away after either touching or leaving a small or larger gap. The velocity was set so that the presentation occurred in less than 300 milliseconds, again preventing eye-movements, and ensuring that peripheral vision was used to make the decision. In Experiments 3 and 4, the direction of motion was randomised, and 3 levels of distractors moving around randomly were presented. In Experiment 3, they were bright red, thus creating a pre-attentive pop-out task, and in Experiment 4, they were the same colour as the colliding entities, thus creating a serial search task. In Experiment 4, we also allowed collisions to occur at the fixation location, so we hid the fixation cross while the stimuli and distractors were being displayed.

The main questions with which we concern ourselves in this paper are as follows: Does eccentricity affect the ability to detect whether a collision anomaly has occurred, and if so which, if any, of the perceptual models of collision perception proposed in Section 3 matches most closely the behaviour observed? In experiments 1, 2 and 3, within groupings of factors, an eccentricity effect was evident, but not very strong. The effects of location and direction of orientation or motion were found to have a weak influence in experiment 1, and a very strong influence on performance in experiments 2 and 3, thus weakening the overall eccentricity effect. The bright red distractors in experiment 3 (the pop-out task) deteriorated overall performance, but the number of distractors did not appear to have an effect.

However, the task which most resembles the real-life situation which we are considering, i.e. the serial search task provided in experiment 4, produced results that exhibited a strong eccentricity effect (see Figure 7). In all experiments, the difference in performance with the larger gap was significantly better than with the smaller one. We can see that the observed behaviour is most closely matched by a function of the form F2, as presented in Section 3, Figure 3. We found in Section 4 that the best reduction in perceptual inaccuracy was achieved by using priority queue scheduling, when inaccuracy was measured using F2 as our perceptual model. This has now been proved to be a valid model in situations where we are animating large numbers of visually similar objects.

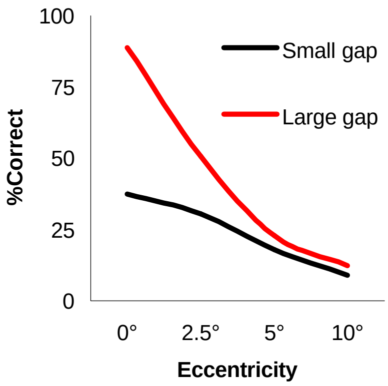


*Figure 7. Eccentricity effect in Experiment 4*

## 7 Conclusions and Future Work

We have presented a model of human visual perception of collisions, and have validated it psychophysically. We have shown the feasibility of using this model as the basis for perceptual scheduling of collisions in a real-time animation of large numbers of homogeneous objects. It has been demonstrated that by using a priority queue scheduling algorithm, perceived inaccuracy can be approximately halved when animating 300 or 500 objects.

However, this model now needs to be refined to make it applicable in more general cases. The results of the psychophysical experiments have demonstrated that other factors, such as location and direction of motion, can have very strong effects under certain circumstances. Further psychophysical experiments into other factors which we have not yet addressed, such as velocity, acceleration, colour and luminance must also be conducted, if the model is to be truly representative of human behaviour.

Much work remains to be done to improve the application also. At the moment, the collision time-step is equal to the rendering time-step, which can lead to objects interpenetrating or tunnelling through each other if the time-step is too large. We are working on adapting the time-step for high-priority collisions, also using the perceptual model. However, it may be that some level of interpenetration must be accepted as a trade-off, so we must also study the perceptual response of the human visual system to this anomaly also. In addition, a more realistic response must be generated, using the laws of physics. The effects on this process of reduced information about points of contact is also being investigated.

Finally, the use of a low-cost, mobile eye-tracker results in a certain amount of spatial and temporal inaccuracy. We need to measure this inaccuracy and incorporate some fault-tolerance into our system. Techniques such as Kalman filters may be useful here. Another possible approach could be to omit the eye-tracker, and to develop a model that predicts where the next likely fixation will be in each frame. Salience maps [18] could be used to achieve this.

## Bibliography

1. Albright, T.D. (1989). Centrifugal directional bias in the middle temporal visual area (MT) of the macaque. Vis. Neurosci. 2, 17, pp 7-188.
2. Aubert, H. Foerster, 0. (1857). Beiträge zur Kentniss des indirekten Sehens (I). Untersuchungen über den Raumsinn der Retina. Arch. Ophthamology, 3, pp 1-37.
3. Barzel, R. Hughes, J.F. Wood, D.N. (1996) Plausible Motion Simulation for Computer Graphics Animation. Computer Animation and Simulation '96. 183-197.
4. Chung, S.T.L. Levi, D.M. Bedell, H.E. (1996) Vernier in Motion: What Accounts for the Threshold Elevation? Vision Research, Vol 36, No. 16, pp. 2395-2410.
5. Cohen, J.D. Lin,M.C. Manocha, D. Ponamgi,M.K.(1995) I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scaled Environments. Proceedings of ACM Int. 3D Graphics Conference. 189-196.
6. Deering, M. (1992). High Resolution Virtual Reality. Computer Graphics, 26, 2, 195-202
7. DeValois, R.L, and DeValois, K.K. (1988). Spatial Vision. New York: Oxford University.
8. Ferwerda, J.A. Pattanaik, S.N. Shirley, P. Greenberg, D.P. (1996) A Model of Visual Adaptation for Realistic Image Synthesis. SIGGRAPH '96. 249-258.
9. Ferwerda, J.A. Pattanaik, S.N. Shirley, P. Greenberg, D.P. (1997) A Model of Visual Masking for Computer Graphics. SIGGRAPH '97. 143-152.
10. Funkhouser,T.A. Sequin, C.H. (1993) Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments. SIGGRAPH '93 247-254.
11. Gottschalk, S. Lin, M.C. Manocha, D. (1996) OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. SIGGRAPH '96.
12. Graziano, M.S.A. Anderson, R.A., Snowden, R.J. (1994) Tuning of MST Neurons to Spiral Motions. Journal of Neuroscience, 14(1): 54-67.
13. Greenberg et al. (1997). A Framework for Realistic Image Synthesis. SIGGRAPH'97. 477-494.
14. Hubbard, P.M. (1995) Collision Detection for Interactive Graphics Applications. IEEE Transactions on Visualization and Computer Graphics. 1(3) 218-230.
15. Hubel, D.H. Wiesel, T.N (1968). Receptive fields and functional architecture of monkey striate cortex. J.Physiology, 195, 215-243,
16. Jacob, R.J.K. (1995) Eye Tracking in Advanced Interface Design. In Barfield,W. and Furness, T.A. (Eds) Virtual Environments and Advanced Interface Design. 258-288.
17. Klosowski, J.T. Held, M. Mitchell, J.S.B. Sowizral, H. Zikan, K. (1998) Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs. IEEE transactions on Visualization and Computer Graphics 4(1).
18. Mahoney,J.V. Ullman,S. (1988) Image chunking defining spatial building blocks for scene analysis. Z.Pylyshyn (Ed.) Computational processes in human vision: An interdisciplinary perspective.
19. Nies, U. Bedenk, B., Heller, D. & Radach, R. (1998). Eye movement patterns during search in a homogenious background. In Becker, W., Deubel, H. and Mergner, T. (Eds). Current Oculomotor Research: Physiological and Psychological Aspects. New York: Plenum Publishers.
20. Palmer, I.J. Grimsdale, R.L.(1995) Collision Detection for Animation using Sphere-Trees. Computer Graphics Forum, 14(2) 105-116
21. Quinlan, S. (1994) Efficient Distance Computation between Non-Convex Object. Proceedings International Conference on Robotics and Automation. 3324-3329.
22. Saarinen, J. (1994) "Visual search for global and local stimulus features" *Perception* **23** 237-243
23. Schiff, W. Detwiler, M.L. (1979) Information used in judging impending collision. Perception 8, 647-658
24. Skerjanc, R. and Pastoor, S. (1997) New generation of 3-D desktop computer interfaces, SPIE Proceedings 3012 - Stereoscopic Displays and Virtual Reality Systems IV (The Engineering Reality of Virtual Reality), 439 - 447
25. Tootell, R.B.H, Silverman, M.S. Switkes, E, De Valois, R.L. (1982). Deoxyglucose analysis of retinotopic organization in primate striate cortex. Science, 218, 902-904.
26. Treisman, A. (1982). Perceptual Grouping and Attention in Visual Search for Features and for Objects. Journal of Experimental Psychology: Human Perception and Performance, 8, 194-214.
27. Weymouth, R.W. (1958). Visual sensory units and the minimal angle of resolution. American Journal of Ophthamology, 46, pp 102-113.
28. Yap, Y.L. Levi, D.M. Klein, S.A. (1987) Peripheral hyperacuity: isoeccentric bisection is better than radial bisection. J. Opt. Soc. Am. A, Vol 4, number 8, pp. 1562-1567.
29. Zeki, S.M. (1974) Functional Organization of a Visual Area in the Posterior Bank of the Superior Temporal Sulcus of the Rhesus Monkey. J. Physiology. 236, 549-573.