Technical Report

UCAM-CL-TR-717
ISSN 1476-2986

Number 717

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Psychologically-based simulation of human behaviour

Stephen Julian Rymill

June 2008

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted 2006 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Jesus College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

The simulation of human behaviour is a key area of computer graphics as there is currently a great demand for animations consisting of virtual human characters, ranging from film special effects to building design. Currently, animated characters can either be laboriously created by hand, or by using an automated system: however, results from the latter may still look artificial and require much further manual work.

The aim of this work is to improve the automated simulation of human behaviour by making use of ideas from psychology research; the ways in which this research has been used are made clear throughout this thesis. It has influenced all aspects of the design:

- Collision avoidance techniques are based on observed practices.
- Actors have simulated vision and attention.
- Actors can be given a variety of moods and emotions to affect their behaviour.

This thesis discusses the benefits of the simulation of attention; this technique recreates the eye movements of each actor, and allows each actor to build up its own mental model of its surroundings. It is this model that the actor then uses in its decisions on how to behave: techniques for collision prediction and collision avoidance are discussed. On top of this basic behaviour, variability is introduced by allowing all actors to have different sets of moods and emotions, which influence all aspects of their behaviour. The real-time 3D simulation created to demonstrate the actors' behaviour is also described.

This thesis demonstrates that the use of techniques based on psychology research leads to a qualitative and quantitative improvement in the simulation of human behaviour; this is shown through a variety of pictures and videos, and by results of numerical experiments and user testing. Results are compared with previous work in the field, and with real human behaviour.

Acknowledgements

First and foremost I would like to thank my supervisor Dr Neil Dodgson for his support, advice and encouragement over the whole course of my PhD. Thanks also to all those in the Rainbow Group, in particular Mark Grundland, who have made suggestions over the years.

Special thanks are also due to all those who were subjected to my interviews and experiments: Margaret, David R., David S., Tom, Rosie, Ross, Jamie, Neil, Peter, Alan, Julian, Richard, Ursula, James, Will, Greg, Daniel, Gareth and Darren.

This work was supported by the Engineering and Physical Sciences Research Council; I am also grateful to the Royal Academy of Engineering, Jesus College, and the Computer Laboratory for their additional financial support.

To all those who helped to proof-read this thesis, many thanks: David S., Tom, James, Luke and Curon, and particularly Ross and my family.

Finally, I would like to thank my parents for all of their support in numerous ways over the years, which has enabled me to be where I am today, and for which I could never thank them enough; and also all my friends for being there when I needed them.

Contents

Chapter 1

Introduction

In recent years the field of 3D animated computer graphics has grown from being an area of solely research interest to being a technology that has revolutionised the entertainment industry. It is hard now to find a film that has not had at least some digital effects added, and the number of completely computer-generated films is increasing. Television programs now have effects that used to be seen only in high-budget films, and computer games contain real-time graphics that not long ago would have taken days to render.

One of the areas of animation that has proved particularly difficult has been the production of animated people, due to the fact that human observers seem to be able to tell very easily if an animated character looks realistic or not; this is no doubt due to our extreme familiarity with seeing other people.

The creation of computer-generated people can be divided into two main challenges: creating accurate, photo-realistic, animated models that look like real people, and then making those models behave in a believable way.

The first of these challenges has received a lot of interest, and this has led to impressive results: motion capture data, recorded from the movements of real actors, can be used to animate 3D models, and improvements in other areas of computer graphics mean that skin, hair and clothes can be reproduced accurately.

However, the second area, that of automatically creating believable behaviour, is still an open topic of research, with the eventual goal being a system in which any number of virtual actors can be placed into a virtual environment, where they will behave in a manner that will fool observers into thinking they are watching a real crowd of people. To accomplish this, the actors must interact with their surroundings, moving through the scene with clear goals and motivations, and interact with each other, avoiding others in a sensible way.

Human behaviour is not only of interest to computer scientists trying to reproduce it: it has been studied for many years by psychologists. Through observations and experiments a large body of literature has been built up, trying to codify the complexities of human behaviour into a set of rules that define how people avoid collisions with others, how they observe their surroundings, and how their behaviour varies based on their mental state.

Given the wealth of material available, it is perhaps surprising that very little use of it has been made in the design of human behaviour simulation systems. Instead, rules for behaviour are generally tweaked on an ad hoc basis until the result looks sufficiently realistic for the required purpose. Throughout this work graphical, quantitative and experimental evidence will be presented to show that making use of psychology research leads to more accurate and realistic simulations.

1.1 Aims

The aims of this work are to improve the simulation of human behaviour by using ideas from psychology research to influence all aspects of the design of the system. This can be divided into three main areas:

1. Collision avoidance behaviour: all techniques for avoiding collisions, and for dealing with collisions if they do happen, should be based on observations of what humans do in these situations.

2. Gaze and attention simulation: the eye movements and attention shifts of each actor should be simulated, and only information acquired about the scene in this way should be available to the actor as it plans its behaviour.

3. Mental state: each actor should be given a configurable set of moods and emotions which will have an effect upon its behaviour and the way it interacts with others.

The aim of this work is *not* to produce a photo-realistic animation: rather, it is to produce an accurate simulation of behaviour.

1.2 Motivations

There are clear uses for an accurate simulation of human behaviour in a number of different areas: architecture, films and games.

1.2.1 Architecture

Most buildings are now visualised with 3D computer graphics during the design process, so that clients can better decide if the design will meet their requirements. This process would be made even more useful if the 3D models could be populated by virtual actors; not only would it make the building look more interesting, but it could help test the suitability of the design. Examples are:

- Different designs could be tested to see which can be evacuated in the shortest possible time, for example in case of fire.
- Bottlenecks in the design could be discovered and eliminated.
- Different options could be compared: should a room have one wide exit or two narrow ones? Is it better to place two exits side by side or on opposite sides of a room?

For the purposes of this type of design testing it is clearly important that the behaviour of the actors is an accurate re-creation of real behaviour, otherwise the results obtained will be useless. It is also helpful if the simulation can be run in real-time, to allow quick testing of new ideas.

1.2.2 Films

Film scripts often call for large numbers of extras, whether for armies, stadiums full of people, or simply crowded streets. Increasingly directors are deciding to cut the costs involved in using people for these roles, and instead creating crowds digitally. These virtual actors have the following requirements:

- they must look believable to an observer;
- the director must have the ability to alter the behaviour of actors to fit the script;
- an observer should be able to empathise with the actors, for example recognising if people are afraid;
- the animation of the actors should look completely natural, even down to their eye movements.

1.2.3 Games

As the power of home computers increases, it is now possible to create crowds in real-time that can react to the actions of the player of a game; for example, crowds of enemies trying to attack the player.

The most important requirement for this type of situation is that the actors can be simulated in real-time, and not have too great an impact on other processing requirements of the game. Currently the need for a photo-realistic appearance is less than for films, but the behaviour must still look believable.

1.3 Structure

This thesis is divided into seven further chapters:

- Chapter 2, *Background and Previous Work*: an introduction to virtual actors, including details of previous work in the field.
- Chapter 3, *Psychology Review*: a review of the relevant parts of psychology literature relating to human behaviour, gaze and attention, and moods and emotions. The emphasis for each source is on relevant aspects of the theories that can actually be implemented as algorithms in a simulation.
- Chapter 4, *Vision and Attention*: this chapter explains how the ideas from psychology discussed previously have been taken and converted into algorithms. Areas covered include control of an actor's eye movements, how the positions of other actors are detected, and how this information is used to build up a mental model of the scene.
- Chapter 5, *Behaviour Simulation*: a description of how attention simulation has been used, together with rules for behaviour derived from psychology, to produce a believable simulation of a crowd of people.
- Chapter 6 *Implementation and Evaluation*: an overview of some of the practical implementation details, followed by evidence of the simulation in action, along with results of qualitative and quantitative evaluation showing the benefits of attention simulation.
- Chapter 7, *Moods and Emotions*: this chapter discusses how the simulation of behaviour can be improved by allowing each actor to be given varying moods and emotions that affect their behaviour and eye movements. Results are presented to show that it is possible to identify an actor's mental state.
- Chapter 8, *Conclusions and Future Work*: an overall evaluation of the work done, and ideas for further work.

These are followed by two appendices, containing programming details and numerical results.

1.4 Definitions

To minimise confusion, certain standardised words will be used:

- An **actor** will mean a virtual actor, a computer-generated character controlled by a set of rules, represented by a 3D animated body. Terms used by other authors include agent, avatar, synthetic actor and non-player character. Actors will be referred to using neuter pronouns.
- A **person** will mean a real, human person, and be referred to with a masculine pronoun.
- A **collidee** is somebody that an actor is predicted to collide with.
- The **subject** is the actor currently being considered.

Chapter 2

Background and Previous Work

This chapter starts with a brief background on virtual actors and the ways they have been used, concentrating principally on their use in films. There then follows a survey of previous research that has been done in this area, looking first at ways of producing accurate animated models, but concentrating particularly on the differing ways of simulating behaviour.

2.1 Virtual actors

The main uses for virtual actors in the past have been in films and interactive virtual environments.

2.1.1 Films

The first ever use of a fully computer-generated virtual actor in a film was a knight created for the film *Young Sherlock Holmes* by Industrial Light and Magic in 1985. The effect was simplistic: the actor was only needed for a scene involving smashing through a window. However, it set the scene for 20 years of improvements that have seen virtual actors take the lead roles in some films.

Other early films that included virtual actors were *Terminator 2* (1991) – for the liquid metal T-1000 robot, *Jurassic Park* (1993) – for many species of dinosaur, *Casper the Friendly Ghost* (1995) – which was the first film with a virtual actor in the lead role, *Jumanji* (1996) – which contained detailed fur and hair effects, and *Starship Troopers* (1997) – whose computer generated characters could be animated in real-time.

Figure 2.1: Crowd scenes from *The Lord of the Rings*

In 1998 *Titanic* was released; this film was notable as it was the first time that virtual actors had been used in large quantities to create simulations of crowds – in this case, virtual crew and passengers walking on pre-determined paths around the ship.

Other notable films are *Star Wars: Episode I* (1999), which contained, for many, the first memorable virtual principal character, and *Final Fantasy* (2001), which was the first completely computer-generated film featuring human virtual actors.

The most recent breakthrough in the use of virtual actors has come in *The Lord of the Rings* (2001–3). Their many uses included: hero characters, such as Gollum and Shelob; digital doubles, to perform stunts that would be too dangerous for real actors to film, or too impractical to shoot; and crowds, such as the armies of orcs, men and elves in the three large battle scenes (seen in figure 2.1).

Finally, *The Polar Express* (2004) was the first feature film where all the virtual actors were produced entirely from motion capture data.

2.1.2 Virtual environments

A virtual environment is, at its simplest, a 3D computer-generated space with which a user can interact. The user may have an immersive experience through the use of virtual reality headsets, or simply see the scene on his monitor.

The idea of virtual reality was first suggested in a novel published in 1964 (*Simulacron-3* by Daniel F. Galouye), which tells of a virtual city whose inhabitants think they are conscious, even though they only exist in a computer. (Some claim that the idea was in fact first suggested over two thousand years earlier by Plato in his *Allegory of the Cave*.)

One of the earliest actual virtual environment systems was the "Aspen Movie Map", created in 1978, which allowed the user to take a virtual tour through the city of Aspen, Colorado; as well as using a collection of photos, there was also a crude polygonal 3D model rendered in real-time.

Early virtual environments were generally single-player: the only human-controlled character would be the user on whose computer the simulation was running. However, there might also be computer-controlled characters, either simply to act as background to make an environment look populated, or to interact in some way with the user; examples of the latter would be characters that would help a new user find his way around, or give him clues in some sort of quest.

In the 1990s multi-player worlds were created, in which dozens of users could participate, in addition to computer-controlled characters; users might well not be in the same physical location, but connected to each other over the internet. Although the most common use for this type of technology is games, in particular role-playing games, there are also educational uses, for example allowing a group of people to explore a re-creation of a historical building, or a virtual museum, perhaps with computer-controlled tour guides.

More recently persistent online worlds have emerged, both Massively Multi-player Online Role-Playing Games (MMORPGs) and general-purpose virtual worlds. The latter of these generally allow users to create a virtual home for themselves, and interact with other users in shared areas; users are able to create customised avatars to represent their desired image. In 2005 it was estimated that 10,000,000 people play MMORPGs and 100,000 participate in general-purpose worlds.

However, there are still current uses for virtual environments in which only a single user participates: VRML and, more recently, X3D allow web users to explore online 3D virtual environments; a particular example is architects modelling buildings in 3D to allow customers to explore them.

It is predicted that the next development in virtual environments will be the integration of many different disparate virtual worlds into a global distributed system, using standardised open-source protocols.

2.2 Previous work

Previous research into virtual actors can be divided into three areas:

- motion: low-level animation of a 3D model;
- appearance: rendering techniques used to give more photo-realistic results;
- behaviour: high-level control of the movements of an actor.

The following sections present highlights of research in all these areas; the selection includes some of the first work in the field, any research which made a particularly useful contribution, and some examples of recent work.

2.2.1 Motion

Three main techniques have been used to simulate the low-level motion of human walking: kinematics, dynamics and motion capture.

2.2.1.1 Kinematics

In direct kinematics an animator sets the position of an articulated human figure at a small number of key-frames, and interpolating techniques are used to work out the in-between positions. The problems are designing suitable key-frames, which requires a skilled animator, and creating adequate interpolations; in particular, it is hard to enforce constraints, such as specifying that the feet should not penetrate the ground. This can be solved using inverse kinematics: in this technique the animator specifies the motion of the end-link in a chain of 'bones' (e.g. a foot); the system then computes the positions of all links in the chain that would produce this result. Clearly the system has to have some biomechanical knowledge of the ways in which a human skeleton can move.

The first use of kinematics to produce animation was by Zeltzer in 1982 [Zel82]. His method used finite state machines controlled by high-level parameters, such as step length and frequency, to automatically generate families of key-frames using biomechanical information. Linear interpolation of these key-frames was then used to produce smooth animation. An alternative technique was described by Bruderlin [BC93], using only direct kinematics, but solving the problem of feet penetrating the ground by always making the support foot the root of the skeleton, and fixing its position; the motion of the support leg is then simulated as an inverted pendulum.

Other work was done by Chung and Hahn [CH99], looking at giving a user different levels of control over motion, and by Wilhelms and van Gelder [WvG97], consisting of a complete simulation of the human body down to the level of bones and muscles. Grochow et al. [GMHP04] describe a system that takes a set of constraints and produces the most likely pose satisfying those constraints, based on an automatically learned model of human poses.

The main problem with using kinematics is that it requires a high quality biomechanical model; in addition, it is very hard to adapt the model to allow walking on non-flat terrain.

2.2.1.2 Dynamics

Dynamics techniques make use of Newton's Laws of Motion, in particular the link between force and resulting motion. Direct dynamics is the application of the laws to calculate the motion generated by a given force, while inverse dynamics calculates the forces that would generate a given motion. This approach is applied to walking by simulating the human skeleton as a hierarchy of rigid solids (limbs) connected by joints.

The first work in this area was by Isaacs and Cohen in 1987 [IC87]. Their approach combines direct and inverse dynamics: the motion of some of the limbs is specified by the animator, from which the system calculates the forces that would cause these motions using inverse dynamics; the motion of the other connected limbs caused by these forces is calculated using direct dynamics. More recently Ko and Badler [KB96] proposed a system whereby motions were calculated using kinematics, but then checked to make sure they were physically possible using dynamics.

A variation on the technique is 'space-time constraints', first proposed by Witkin and Kass [WK88]. In this the user sets constraints on initial, final and some intermediary positions and velocities; the system imposes constraints that limit muscular forces, and ensures that Newton's laws are followed.

Work was done on animating human athletics by Hodgins et al. [HWBO95], on interactive animation based on an underlying physical model by Laszlo et al. [LvdPF00], and on a new framework for physically-based animation by Faloutsos et al. [FvdPT01].

Fang and Pollard [FP03] attempted to speed up the process of calculating constraints on poses by only making use of first derivatives. Liu, Hertzmann and Popovic [LHP05] used a non-linear inverse optimisation algorithm to extract biological information, such as joint stiffness, from motion capture data, which is used by a dynamical model to create new motion. Zordan et al. [ZMCF05] use a physical simulation, which responds to contact forces between the actor and other objects, to fill the gaps in between existing sequences of motion capture data.

2.2.1.3 Motion capture

Kinematics and dynamics approaches were popular during the 1980s and early 1990s. However, both suffer from the difficulty of trying to model the extremely complex ways in which real people walk; results from these techniques always look slightly artificial. However, in the mid-1990s a radically different technique was developed, known as motion capture, which makes use of actual data about human motion. The motion is captured from a person wearing a special suit covered in optical or magnetic markers; from these position and orientation data can be recorded, and an animated skeleton produced. Kinematics techniques can then be used to re-target the motion onto a skeleton of different proportions.

Early work on motion capture was carried out by Unuma [UAT95], who described ways of starting with a database of characteristic human motions and using Fourier expansions to interpolate and extrapolate their parameters to produce new motions. Witkin and Popovic [WP95] proposed an alternative technique known as motion warping, using constrained interpolation so that the resulting motion will fit specified key-frames but preserve features of the original motion.

Gleicher [Gle98] looked at ways of adapting animation to fit a different character, Popovic and Witkin [PW99] continued to look at ways to generate new animation sequences using space-time constraints (e.g. footprints), and Metaxas [Met01] outlined ways to adapt motion capture data to produce motion along curved paths and over uneven terrain. Polichroniadis [Pol00] created an animation framework which used forward kinematics for low-level control of joints, but transformed motion capture data for high-level positioning of limbs; he also considered ways of blending together pieces of motion capture data.

Two other novel related techniques were investigated: using simple, stylised animation provided by the animator to create more accurate motion [LP02], and applying 2D motion captured from cartoons to a 3D model [BLCD02].

Work by Arikan, Forsyth and O'Brien [AFO03] allows a user to describe a timeline for the motion of an actor using annotations ('run', 'jump'), which are then synthesised from motion capture data. Safonova, Hodgins and Pollard [SHP04] describe a system in which the user can specify motion in a sparse, intuitive way, and existing motion capture data is used to create a low-dimensional subspace in which the specified motion can be optimised.

Anguelov et al. [ASK+05] look at ways of producing whole-body animation from sparse motion capture data by re-creating realistic muscle deformation. Chai and Hodgins [CH05] describe a way of simplifying the motion capture process: users only wear a small number of reflectors, and the system uses a library of existing motion data to fill in values for the information about the user's motion that was not captured by the reflectors.

2.2.2 Appearance

Previous work on the appearance of virtual actors has looked at ways of simulating skin as it moves over the bones of a skeleton, ways of simulating hair

Figure 2.2: Gollum, from *The Lord of the Rings*

and clothes, and ways of creating lifelike facial expressions. Given that these areas are not considered elsewhere in this work, details will not be given here. However, it is worth noting in passing that the work by Weta Digital on the character Gollum (figure 2.2) for *The Lord of the Rings* typifies the current state of the art: subsurface scattering to create more realistic skin tones, simulation of the movement of every single strand of hair, new reflection techniques for the eyes, and facial expressions – created in part from motion capture – detailed enough to allow the emotions of the character to be seen clearly (and, in fact, to allow lip-reading) [For03].

2.2.3 Behaviour

Given that the subject of this work is the simulation of human behaviour, an extensive survey of previous work in this area is presented, starting with early work on animal behaviour, and then looking at different ways of simulating human behaviour.

2.2.3.1 Animal behaviour

The first work to be done on the computer simulation of behaviour was conducted by Reynolds [Rey87], who produced a set of algorithms to simulate flocking behaviour in birds (shown in figure 2.3). Each 'boid' was given three rules: move towards nearby boids, try to match speed with nearby boids, and

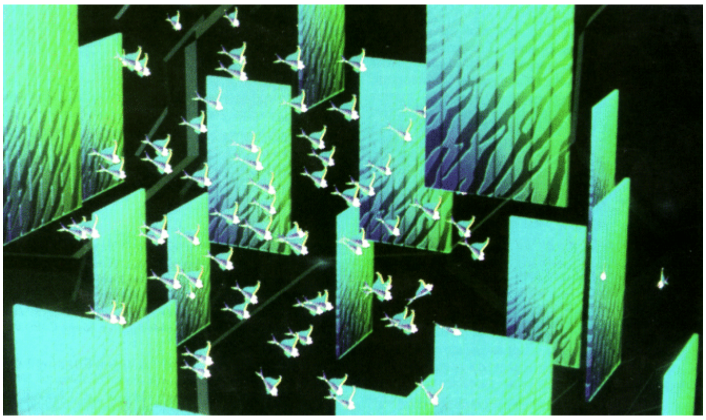

Figure 2.3: Reynolds' boids system [Rey87]

avoid collisions. His real breakthrough was in realising that the combination of every boid in the flock having these simple rules would result in an overall complex behaviour, as opposed to the alternative approach of trying to write by hand the complicated rules that govern the behaviour of the entire flock.

Further work on animal behaviour was done by Tu and Terzopoulos [TT94], who described a very detailed simulation of fish, using a similar system to Reynolds', also including a complicated system to deal with goals, giving each fish a mental state and using this to work out its intentions.

A somewhat different approach was taken by Sims [Sim94], who produced virtual creatures whose appearance and motions are evolved through mutations. The evolution can be directed using fitness evaluation functions so that desired goals such as walking may be achieved. This technique has the benefit of not requiring the programmer to design the behaviour of the creature; instead a user can observe what behaviours are being evolved and choose the best one.

Reynolds continued his work [Rey99], building on his original model to give an implementation of some simple steering behaviours, such as seek, pursuit, and obstacle avoidance; combined together these can produce fairly complex behaviour.

There are a great many other papers on animal behaviour, for example: a virtual pets system [GC98], a predator-prey model [NI97], and another bird simulation [LW93]; on the whole, most work is simply an adaptation of the general boids model to a particular situation.

Finally, Anderson, McDaniel and Chenney [AMC03] described a method for imposing hard constraints on the paths of boids at specific times, while retaining the global characteristics of a flock, thereby allowing more control by the director.

2.2.3.2 Agents

The term 'agent' is often used to refer to virtual actors, or virtual creatures, particularly in systems which give the animator multiple levels of control over the behaviour of the agent.

The term is demonstrated well by Blumberg and Galyean [BG95], who introduced a layered system allowing control of agents at multiple levels. The director can specify exact movements (e.g. move leg), tasks (e.g. find food), or motivations (e.g. hunger). The advantage of this system is that it can function completely autonomously if desired, but can also be controlled at any level of detail; this is useful if the system is being used for special effects in a film, as it allows the animator to ensure that the actors perform the required movements (e.g. moving across the scene), while still doing so in an autonomous manner.

Badler et al. [BAZB02] tried to parameterise agent behaviour by presenting variables that control all aspects of behaviour at the same time; this ensures that behaviour is consistent across the body, in speech, facial expression, eye movement, head movement, limb gestures, body posture and gait. The system accepts input through a natural language, and uses this to set the emotion of the actor.

The agent approach is also used by a large quantity of the work discussed in section 2.2.3.4.

2.2.3.3 Navigation

Actors in a simulation will normally be given goals to walk towards, either just to create movement that looks interesting, or because the goal is an integral part of the simulation; for example, the goal might be to evacuate a building, or to get across a road. To make progress towards this goal the actor must plan the route it is going to take. Research into route-planning has been conducted, both in the fields of virtual actors and physical robot navigation.

Franz and Mallot [FM98] published a useful survey of different techniques, listing a hierarchy of increasingly complicated goal-finding techniques: search (random movement until goal is found), direction-following (travel in known direction towards goal), aiming (always point at goal), guidance (find goal based on relation to surroundings), recognition-triggered response (following a previously learnt route), topological navigation (combining routes by knowing that they pass through the same place), and survey navigation (finding paths over novel terrain). Each of these builds on techniques from lower down in the hierarchy.

Trullier et al. [TWBM97] describe a further technique, known as metric navigation, which uses metrics about the world, such as distances and angles, to calculate shortcuts and detours from a planned route.

Kuffner [Kuf98] proposed a novel technique where a 2D overhead view of the scene is created, using 3D graphics hardware, and then Dijkstra's algorithm ([Dij59]) is used to find the shortest route.

A paper by Bandi and Cavazza [BC99] is one example of the use of the A* search algorithm ([HNR68]) for route-planning by actors; A* search is the most computationally efficient algorithm that is guaranteed to find the shortest path. Of course, this technique assumes that actors have a correct internal map of their surroundings.

Other work has been by Pettre, Laumond and Thalmann [PLT05], who use a navigation graph to plot paths over multi-layered and uneven terrains; they also [PT05] investigated ways of producing a variety of paths between different locations, rather than just the shortest route. Chenney [Che04] introduced the idea of 'flow tiles', which define velocity fields that can then be applied to actors, carrying them towards their goals. Brogan and Johnson [BJ03] used pedestrian performance statistics that were obtained during a set of experiments to create a path-planning algorithm that more closely matches real behaviour.

2.2.3.4 Human behaviour

The route planning techniques described in the previous section will allow an actor to avoid large-scale obstacles, such as walls or roads. However, they do not help with the avoidance of small obstacles, such as banana skins, or, more importantly, other actors. To allow actors to avoid these types of collisions they must each be given a set of behavioural rules to follow; research into the design of these rules is now presented.

Early work on human behaviour simulation was conducted within the fire safety community by Thompson and Marchant [TM95], who created a system to test how quickly different designs of buildings could be evacuated. There were a number of over-simplifications in the system: the speed of an actor is determined just from its distance from other actors (so an actor will slow down even if the nearest actor is behind it); actors will always overtake if they can increase their speed by doing so, and will always overtake on the side that deviates less from their chosen path; actors have access to a fully accurate map of the buildings, and compute the optimal way out.

In the same year, Helbing and Molnar [HM95] proposed a 'social force' model to predict how people should move, based on the motivations of the individual and on the positions of other actors. The result is an extremely complicated mathematical approach, but one which does successfully model lane formation among pedestrians and turn-taking effects when passing through narrow openings; however, this seems to be due to the fact that these are explicitly coded in the model. One strange claim is that people do not like walking near walls, in case they might collide with them; this would seem to have no basis in reality.

One approach to producing more efficient code is to simulate groups of people rather than individuals. Work by Thalmann, Musse and Kallmann [TMK00] collects actors together if they share a common goal and controls them as one group. In common with other systems, this makes use of a hierarchy of levels of control: a group of actors can be guided (under the full control of the user), programmed (route supplied by user) or autonomous (finds own route). The problem with this approach is that even if people share the same goal they may still behave in very different ways, such as in their speed of walking and in their interactions with others. Earlier work by Musse and Thalmann [MT97] again defines a crowd as a group of people who share a common goal. Its multiresolution collision avoidance scheme is worthy of mention: proper collision detection is only done for actors near the camera; those further away are ignored. This seems to somewhat miss the point, as even if the viewer cannot see the collisions they will still have an effect on the paths of the actors; for example, with no collision checking there would be no congestion in a crowded area as all the actors would just keep moving through each other at full speed.

Still [Sti00] conducted work on crowd dynamics, with a design based on observations of real crowds at football stadiums. This approach simulates the crowd as an emergent phenomenon using simulated annealing and mobile cellular automata. Feurtey [Feu00] suggested a 'space-time' approach: each actor is given (x, y, t) coordinates; to work out a suitable detour a disc (centred on the actor) of 'admissible displacements' is produced, with other actors causing forbidden areas on the disc. One problem with the system is that it can only consider a maximum of 10 actors interacting simultaneously. In the same year Blue and Adler [BA00] published a cellular automata model for pedestrians. This produces fairly good results, but has the restriction that actors are limited to a square grid and can only move simply from square to square, either advancing in a straight line or moving sideways to change lane. Monzani et al. [MCT01] present a system which gives each actor a set of beliefs and internal states, goals to achieve, and plans: these specify a sequence of actions needed to achieve a specific goal. However, the main focus of the paper is on ways in which actors can interact with 'smart objects' – objects which 'tell' the actor how to use them.

Notable recent work was carried out by Lamarche and Donikian [LD04]; their main area of interest was path-finding, and they present a way of topologically structuring the geometry of the environment to allow a path to be found quickly. However, they also consider pedestrians' reactive behaviour, with each actor being given a configurable navigation model.

Sung, Gleicher and Chenney [SGC04] presented an approach designed to improve scalability: actors normally have very simple rules, but as they enter new situations they are given additional, situation-specific behaviours on the fly. Lai, Chenney and Fan [LCF05] introduced the idea of group motion graphs, which record the motion of a group in a pre-production stage and play it back at run-time; the main issue discussed is how to blend these clips together. Ratner and Brogan [RB05] studied how the dynamics of human walking influence the behaviour of pedestrians, by modelling people as inverted pendulums and considering balance dynamics. Pelechano et al. [POSB05] described an architecture for integrating a psychological model into a crowd simulation system, and considered ideas of communication between actors. Finally, Treuille, Cooper and Popovic [TCP06] presented an approach which does not consider each agent individually, but applies a 'continuum perspective', creating a set of dynamic velocity fields over the domain that guide all the actors simultaneously.

Also of interest is the commercial product Massive [Mas03], created by Weta Digital for *The Lord of the Rings*. It has been designed to be able to simulate crowds of hundreds of thousands of actors, and uses a neural network combined with fuzzy logic. The brain of each actor consists of up to 2000 nodes (shown in figure 2.4), and takes as input the data from sensors re-creating eyes and ears; the output is a command to the rendering engine specifying which piece of motion capture data to use. While producing impressive results, the creator of Massive is quick to point out that it is not an artificial intelligence system; the main contribution is being able to handle such a large number of actors at the same time, all using motion capture data. Anecdotal evidence suggests that Massive simulations need to be carefully set up, and that the simulation can only run for about thirty seconds before unrealistic or unwanted behaviour starts to become obvious.

The work cited here is a selection of the most important research carried out in this area, but much more exists [SSV$^+$99] [dLL99] [CBC01] [Tha01] [VP01] [UCT04] [ST05].

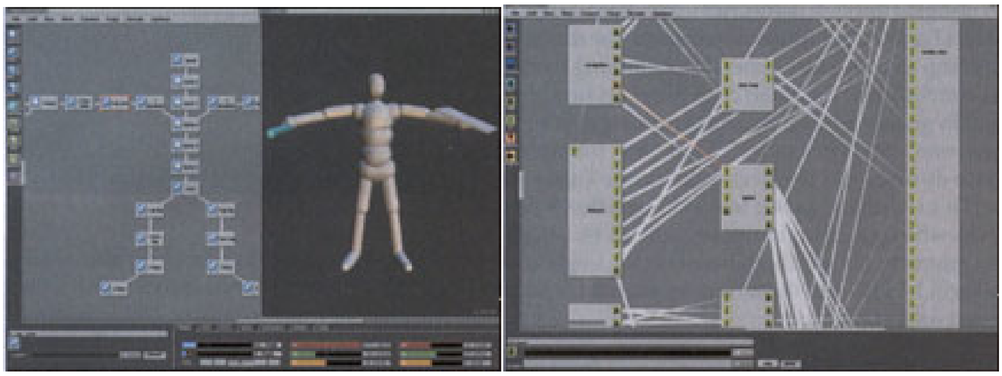

Figure 2.4: The nodes that make up the brain of each actor in Massive [Mas03]

2.2.3.5 Vision simulation

Previous work on simulating vision falls into two areas: database search and computer vision methods.

Database search In this method an actor is allowed to directly interrogate information from the model of the scene to find out the locations of other actors. In its simplest form this would allow actors to have total awareness of all other actors around them. However, there are simple ways of cutting down the information available to give a more realistic simulation. The boids in Reynolds' model [Rey87] can only get information about the positions and velocities of other boids within a fixed radius of their position (although Reynolds states that he would prefer a more accurate system). In later work, Reynolds [Rey00] used a bin-lattice spatial subdivision technique to reduce the number of tests required to determine which boids need to be considered.

The fish in Tu and Terzopoulos' system [TT94] can detect any object that is not fully occluded by another object, and is within a certain distance and a fixed field of view in front of the fish. Bordeux et al. [BBT99] propose a framework involving a series of filters that are applied to data about objects in the world; only objects that pass through all the filters can then be seen by the actor. In their example system these filters are simple re-creations of an actor's field of view.

Most of the human behaviour simulations discussed in section 2.2.3.4 use one of these techniques, but make no real attempts to improve upon them.

Computer vision An example of a computer vision system is found in Massive [Mas03], which uses a method known as image-based vision: this renders the field of view in front of the actor using the graphics card, and then uses computer vision techniques to detect the positions of other actors. This is normally a hard problem in computer vision, particularly due to the varying appearances of different people; however, as the actors are already being drawn by the system, optimisations are possible, as it should know what it is looking for. The technique remains complex, but it solves problems such as occlusion without any additional work. However, the field of view is still fixed to be directly in front of the actor, and it has total awareness within this region, which is not an accurate representation of real human vision.

There has been a small amount of other work in this field: instead of only rendering an image, Noser et al. [NRTMT95] also render two buffers: a z-buffer containing the distance between the actor and the object, and an object buffer containing an identifier for the object at that point in the scene. The use of the z-buffer solves the occlusion problem, so any object in the object buffer should be visible to the actor (even if only in part). Peters and O'Sullivan [PO02] used a false-colouring approach to render the scene from the point of view of the actor, simplifying the process of identifying others. They use a two-level vision process: distinct vision mode, where each object can be identified separately, and grouped vision mode, in which only groups can be detected. As described further in later work [PO03], these two modes are used to re-create the effect of accurate vision in the fovea and limited peripheral vision respectively.

2.2.3.6 Attention simulation

Relatively little work has been done in the area of attention simulation – trying to predict what area of the scene an actor should be interested in, and re-creating the effects upon behaviour that this will have. Preliminary work was done by Osberger and Maeder [OM98], who produced algorithms to automatically identify perceptually important regions in an image, based on factors such as size, motion, and type of object; for example, people are inclined to look at other people, particularly if they are moving. A system was created by Chopra-Kuller [Cho98] where a user could input a command to an actor, such as an instruction to walk towards a table. Appropriate motions would then be synthesised, along with suitable eye movements to show the attention of the actor. The system was fairly simple, in that the target of attention would nearly always be the object which the actor had been told to move towards. Simulations only consisted of one actor, so there could never be any observations of other actors.

Chopra-Kuller and Badler [CKB99] continued to look at ways of producing gaze behaviour in virtual actors, with a system of behavioural agents controlled by finite state machines. Each agent has a simple queue of targets that it needs to look at, which is processed in order. This can cause problems if high priority requests, such as those to look at an actor with whom a collision might be about to happen, are not processed in time. In addition, if an actor wants to monitor another actor, it has to continually re-queue it; this seems inefficient as monitoring is a very common activity.

Further work on attention simulation was done by Hill [Hil99]. He described a system for a synthetic helicopter pilot whose job is to look for enemy vehicles: it is clearly important to simulate where the pilot is currently looking. One of the key areas studied here was the grouping of vehicles into larger entities that can be attended to at the same time; this can either be done automatically, or under the simulated control of the pilot. This work was then continued [HKG02], using a neural network to decide which parts of the terrain should be attended to while tracking vehicles. The system predicts where a vehicle should be, based on where it was when last seen, and its velocity at the time, and that predicted position is used as a starting point for a search for the current position of the vehicle.

Itti, Koch and Niebur [IKN98] [IK99] [IK00] describe an implementation of a pre-attentive selection mechanism; various maps of a scene, containing such information as colour and intensity, are combined together into a single saliency map which encodes the conspicuity of objects in the visual environment. Such a map would be used by a top-down mechanism, with variable selection criteria, to direct the spotlight of attention.

Yee, Pattanaik and Greenberg [YPG01] use a model of visual attention to speed up the process of global illumination by constructing a spatiotemporal error tolerance map of areas in the scene where less work needs to be done. Peters and O'Sullivan [PO02] described a three-stage memory system in which details of objects seen by the actor pass through short-term sensory storage to short-term memory; only a small number of objects can be kept in short-term memory at the same time. The final stage is long-term memory, which contains details of the task the actor is performing. Further work [PO03] describes how this system is used to generate gaze animations such as glance, look and stare. Peters at al. [PPB+05] describe a model which allows a conversational agent to determine the level of attention and interest of a user by monitoring their gaze movements.

Finally, work was done by Gillies and Dodgson [GD05] on attention-based obstacle avoidance. If an obstacle might need to be avoided a request will be made to look at it, and it will then be intermittently monitored. Actors will also undertake spontaneous looking, if nothing is being attended to, searching their environment for items of interest. The result of this is a simulation in which the actor has believable eye movements. However, as with other work, observations of other actors are not considered.

Related work was carried out by Lee et al. [LBB02], looking to create accurate eye movements for an animated face, for use in a conversational agent. They implement an eye movement model based on empirical models of saccades and statistical models of eye-tracking data, and demonstrate that this looks more natural than either stationary eyes or random movements. Naturally their work is focused on the social interaction aspects of gaze, such as looking at somebody to indicate it is their turn to speak, rather than on how eyes move to study a scene.

2.2.3.7 Moods and emotions

Not much work has been done previously on the effects of moods and emotions on the behaviour of virtual actors, and there has been a focus almost exclusively on the effects of mood on non-verbal communication between actors.

Early work on non-verbal communication can be divided into two areas: gaze and posture. The direction of gaze of a person during a conversation will vary based on his mood: the more he looks at the person who is speaking, the more interested he is (or is pretending to be) in what is being said; this effect was simulated by Garau et al. [GSBS01] and by Colburn et al. [CCD00]. Variation of posture based on mood was studied by Coulson [Cou02], and by Becheiraz and Thalmann [BT96]; they cover postures such as arm crossing, leaning forward or back, and turning towards or away from the other actor. These two areas have been combined in work by Gillies and Ballin [GB04], who present a system in which the user can alter the mood settings of an actor (such as affiliation and status); suitable postures and gaze patterns will then be generated.

Other work in this area was by Slater, Pertaub and Steed [SPS99], who conducted an experiment in which participants were asked to give a talk to an audience of virtual actors; the actors could either appear positive – attentive, well-behaved and interested – or negative – hostile and disinterested. The effects of these reactions on the speaker were then measured through questionnaires. The actors expressed their feelings through facial expressions, clapping, nodding, yawning, turning away or walking out.

Polichroniadis [Pol00] implemented a set of signal processing operations that transform pieces of movement acted in a neutral manner into movement that portrays an individual style, expressing emotion or personality traits such as happy, drunk, feminine or confident. He looked at techniques which would allow the user to specify character attributes easily, but which still allow access to the whole gamut of movement styles. His approach was to provide the user with an interface to broadly define the style using adjectives, then explore the range of perceptually similar alternatives.

Brand and Hertzmann [BH00] approached the problem of stylistic motion synthesis by learning motion patterns from a set of motion capture sequences. The model makes use of style machines, which can interpolate between different styles and extrapolate new motions. Neff and Fiume [NF04] consider how the postures that a character adopts expresses its emotions; they balance three components within an inverse kinematics system: meeting world space constraints, finding a body shape that reflects the character's inner state and personality, and maintaining realism through balance. Egges and Magnenat-Thalmann [EMT05] describe an animation system that allows for the synthesis of realistic communicative body motions according to an emotional state: animations are only specified for a few joints, and dependent joint motions are calculated in real-time. Motions are constructed as a blend of idle motions and gesture animations.

Ochs et al. [ONPS05] proposed an architecture for an embodied conversational agent; they present a formalisation of emotion-eliciting events based on a model of the agent's mental state. Fuzzy logic is then used to blend emotions together to produce facial expressions. Additional work by Pelachaud [Pel05] looked at ways of giving each agent individual characteristics, through a set of six parameters which encode gesture and face expressivity. Finally, Hartmann, Mancini and Pelachaud [HMP05] studied ways of modifying a generic gesture to carry a desired expressive content while still retaining its original semantics.

There appears to be no previous work on the effect of moods and emotions upon either walking behaviour or its accompanying eye-gaze movements.

2.3 Summary

This chapter began with a brief introduction to the concept of virtual actors, and then surveyed a number of highlights from a large spectrum of previous research. Of most interest for the present work was the research into human behaviour simulation, and into ways of re-creating human vision and attention.

Chapter 3

Psychology Review

As stated in section 1.1, the aim of this research is to base all aspects of this simulation of behaviour on ideas from psychology research. Therefore, this chapter contains a survey of the most important pieces of research that have been used, divided into three areas: general pedestrian behaviour, vision and attention, and moods and emotions. Sources have been selected and rated primarily on the ease with which the theories expressed can be implemented as algorithms, rather than on their importance and usefulness to other psychologists.

3.1 Pedestrian behaviour

3.1.1 Sources

The sources which proved to be most useful were:

- *Relations in Public* by Erving Goffman [Gof72]
- *Notes on the behaviour of pedestrians* by Michael Wolff [Wol73]
- *Patterns of public behaviour: Collision avoidance on a pedestrian crossing* by Peter Collett and Peter Marsh [CM81]
- *Walking, crossing streets, and choosing pedestrian routes: a survey of recent insights from the social/behavioral sciences* by Michael Hill [Hil84]

Goffman This work is one of several by Goffman studying many different aspects of human behaviour. This book focuses in particular on the way that people behave when they are interacting with others, and so considers areas like personal space, interpersonal relations (such as greetings), etiquette, conversations, and tie-signs (how people give clues to relationships). However, the chapter of interest for this work is a description of pedestrians as 'vehicular units', which gives a set of rules for how people walk along pavements, in particular how they avoid other people. Almost more importantly, he also explains the reasons why people behave the way they do. Unfortunately, most of his statements are just given as assertions, and it is hard to tell whether he is using results from other authors, reporting results of observations, or just proposing a theory. However, this should not be too great a problem, as he is a highly respected author, and this is seen as a seminal text in the field.

Wolff Michael Wolff's paper is the result of observations carried out on a typical city block in New York. He describes the area as a busy thoroughfare, especially congested during the rush hour, populated by a full range of individuals: shoppers, office workers, students, executives, messengers and tourists. The experimenter would select an unaccompanied adult, approach that person on a 'collision course', and see what action they would take to avoid the collision. From these experiments he was able to find the distance at which people started to avoid a collision, how they avoided the collision, what people do after avoiding the collision, and how clusters of people are formed. He also presents some investigations into the behaviour of people in groups, and the difference this makes to avoidance behaviour. However, the most important contribution is the description of a behaviour known as 'step-and-slide', which Wolff was seemingly the first to document.

Collett and Marsh Collett and Marsh also base their work on observations. However, they found that observing pedestrians on the pavement was unsatisfactory as, in most normal situations, the density of people was so low that most of the behaviour was uninteresting. Therefore they carried out observations at a traffic light-controlled pedestrian crossing at Oxford Circus, London, watching as lines of pedestrians on either side of the road were forced to find a way past each other as they crossed. From this large-scale behaviour they picked out the details of one-on-one collision avoidance behaviour, particularly looking at the cases where avoidance came late on. Results are presented on which side people prefer to pass each other, along with more details of the step-and-slide behaviour.

Hill Hill's book is a survey of past work in the field, rather than new research by the author. Inevitably the primary sources are the three mentioned above, but many others are also cited. There are two main areas relevant to this work in the book: one chapter on general walking, and another on crossing streets.

Details of the results discussed in each of the above sources, and others, will now be presented, divided into sections based on the type of behaviour being described.

3.1.2 General pedestrian behaviour

When large numbers of pedestrians are all walking through an area, one of the most surprising aspects of their behaviour is that it is *orderly*: on the whole people do not push and shove each other out of the way, or bump into each other, or try to walk at top speed to get to their destination. A good description of this social order is given by Ross [Ros08]:

> Order cannot be said to prevail among people going in the same direction at the same pace, because there is no [potential for] interference. It does not exist when people are constantly colliding with each other. But when all who meet or overtake one another in crowded places take the time and pains needed to avoid collision, the throng is orderly.

Goffman notes that there are many techniques used by pedestrians to avoid bumping into one another, which are constantly in use and shape patterns of street behaviour; without them, "street traffic would be a shambles". However, most people do not normally give much conscious thought to the techniques they use to avoid others.

Goffman further highlights a few differences between pedestrian traffic and road traffic, which are often thought to be similar. Unlike cars, pedestrians' main goal is often not simply to get from one point to another: they may be shopping, or strolling and chatting. In addition, unintentional contact is far less serious for people than it is for cars: pedestrians can "twist, duck, bend, and turn sharply", and so extricate themselves from a collision at the last moment; even if there is an actual collision there will usually be little damage, unlike vehicle collisions.

Several studies have looked at the average speed that pedestrians walk at. Daamen and Hoogendoorn [DH03] found that the average walking speed when not hindered by others was 1.58 m/s; the minimum was 0.86 m/s and the maximum was 2.18 m/s. However, studies carried out by Knoblauch, Pietrucha and Nitzburg [KPN96] showed that the average speed varied based upon certain factors:

- Men tend to walk faster than women, at 1.56 m/s compared to 1.46 m/s.

- People on their own will walk faster (1.54 m/s) than people in groups (1.42 m/s).
- People will walk faster when it is raining (1.60 m/s) than when it is dry (1.46 m/s).
- People walk more quickly when it is cold (1.60 m/s) than when it is hot (1.48 m/s).

The techniques people use to avoid collisions will now be discussed, split into the two major types of collision, followed by a description of behaviour after a collision has been successfully avoided.

3.1.3 Avoiding oncoming people

An area of particular interest is the way in which people attempt to avoid collisions with others who are walking towards them, such that if no action was taken there would be a head-on collision. This situation is covered by all three principal sources. In the following paragraphs 'subject' will refer to the person whose behaviour is being studied, and 'collidee' will mean the person with whom they would collide if no action was taken.

Wolff carried out experiments to see at what distance the subject will start to avoid the collidee, and how this varies based on the number of nearby people. In a low-density situation, with between 5 and 15 people in the area (15 m by 7 m in size), the median distance at which the subject yielded to the oncoming collidee was 2 m. In higher densities, with between 16 and 30 people in the area, the subject would yield at a distance of 1.5 m. However, Wolff explains that the low-density figure is in fact inaccurate: because of limitations of the camera equipment being used to record the experiments the experimenter would only move onto a collision course with the subject when they were both in camera range; it is suggested that this would make the subject less likely to yield early on, as they would feel they had a right to be on that path, having been on it first. Separate observations backed this up: in low-density situations people would actually often start avoidance behaviour at a distance of up to 15 m away, and possibly even 30 m.

Goffman notes that the reason for avoiding people at such a great distance if possible is that by doing so the subject can make a small angular correction in velocity which will cause a large change in position; this is seen to be less demeaning than having to make a large, sudden movement at the last moment.

Wolff gives a number of possible reasons for the short avoidance distance in high-density situations:

- The subject may not see the collidee until that point due to intervening people.
- The subject may be unsure that the collidee will continue on his collision course, as minor direction changes are common.
- It may be hard for the subject to change direction due to other people close by.
- The subject will be studying his immediate environment as a priority, in case collisions may happen with those people.

Wolff then explains how people actually avoid the collision; once again, behaviour varies based on the density of people in the area. In low-density traffic, subjects will prefer to change their paths (detour) to avoid the collidee; in high-density traffic a more common behaviour is not a total detour but "a slight angling of the body, a turning of the shoulder, and an almost imperceptible side-step", which Wolff calls a 'step-and-slide'.

Both of these forms of collision avoidance are now discussed, along with the consequences of these techniques failing.

3.1.3.1 Detours

Collett and Marsh describe how the subject chooses on which side to pass the collidee: if there is an overlap between the courses of the two people, i.e. their shoulders are overlapping, they will always pass on the opposite, shorter side. For example, if people's right shoulders are overlapping, they will pass to the left of each other, as shown in figure 3.1. The amount of overlap seems unimportant: even if the overlap is near total, they will still pass on the shorter side. In their experiments, not one pair moved in the unexpected direction. The reason given for this is that this minimises the effort required for avoidance to take place.

However, if there is a total overlap between the two people, where people are approaching each other head-on, Goffman notes that they tend to veer to the right, i.e. pass on the right-hand side of each other. His explanation for this is that people tend to follow the same rules that they would while driving, as the study was carried out in Canada where one drives on the right side of the road. However, Collett and Marsh report that worldwide there is a tendency to move to the right. No real reason for this is given, but there is a suggestion

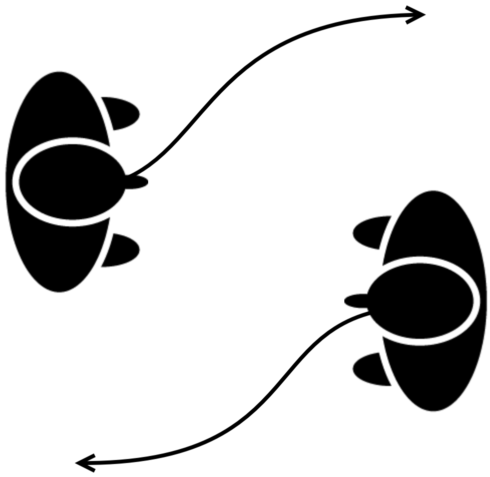

Figure 3.1: If people's right shoulders are overlapping, they will pass on the left of each other

that it is due to most people being right-handed. Alternatively, it may just be a convention: "an individual may conform to the convention, consciously or unconsciously anticipating that the other person, whom he takes to be similarly disposed, will do likewise." In their experiments twice as many people moved to the right as to the left of each other: while a convention clearly exists, it is not an absolute rule. (Collett and Marsh note that an exception to the rule exists in Australia and New Zealand, where people tend to move left – this is explained by people being encouraged to walk on the left of the pavement so as to face oncoming traffic.)

Finally, Collett and Marsh also mention that an alternative to moving right is to "take the initiative and move with sufficient resoluteness both to obviate the need for decision on the part of the other and to discourage him from not cooperating."

Wolff notes in passing that if a subject is approaching a couple, or a small group of people, he will be expected to detour around the entire group; they will not, normally, yield to him. People with bags and packages, and the elderly, are also less likely to assist in the detour.

Goffman describes a way in which people assist others to determine their intentions, in particular on which side they are going to detour, which he calls 'externalisation' or 'body gloss'. The theory is that people use "all-over body gestures to make otherwise unavailable facts about their situation gleanable", conducting themselves "so that the direction, rate and resoluteness of their proposed course will be readable." It is unclear exactly what form these gestures take, but it seems likely that they consist of exaggerated versions of normal

Figure 3.2: The step-and-slide manoeuvre: an angling of the body, a turning of the shoulder, and a side-step

movements, such as deliberately turning the body in the direction of the detour. If others can read these intentions then they can adapt to the chosen course early on without any loss of self-respect. Goffman adds that if two people are on a collision course then they may, having each externalised their intentions, make eye contact to ensure that both of them understand what the other is planning.

3.1.3.2 Step-and-slide

An example of the step-and-slide manoeuvre is shown in figure 3.2. Wolff gives more details: the manoeuvre starts when the distance between subject and collidee is about 1.5 m, at which point if either of the two walked directly ahead there would be a collision at the next step; this is because they have not fully moved out of the path of each other. To avoid a collision happening, *both* people have to cooperate and perform the step-and-slide. However, even with both people executing the manoeuvre correctly Wolff found that in almost all his observations there was still some body contact, such as brushing of shoulders, chest or arms. People tend to pull their hands inward or away to avoid contact, and twist their bodies backwards to maximise face-to-face distance. He also found that if the collidee did not cooperate in the step-and-slide, and contact ensued, the subject would be offended and stare at the collidee, or make remarks such as “Look where you’re going!”.

Collett and Marsh observed that there are two types of step-and-slide: ‘open’, where the subject turns towards the collidee, and ‘closed’, where the subject turns away. Men seem to prefer the open pass, and women the closed pass.

Hill notes that even though people do nearly always cooperate with each other to avoid collisions, they are still irritated by the delay caused by taking avoiding action, particularly detours. This would explain why the step-and-slide manoeuvre is popular, as it requires no deviation from the current path, and therefore does not cause a delay. It is also sometimes necessary if a collision has not been predicted until late on, in which cases if it is not used then a collision would happen, as there would be no time to make a suitable detour.

3.1.3.3 Collisions

Collett and Marsh note that if any one of the processes that usually ensures collision avoidance breaks down, a collision may result or, if the collision is noticed at the last moment, there may be a ‘pedestrian dance’ or a ‘stutter-step’. Sadly, neither of these terms is defined, but they seem to indicate a number of side-steps as people try to get round each other; it may also refer to the situation where people continually try to move the same way and therefore cannot pass.

Hill mentions that there are in fact a very high number of slight collisions (or ‘brushes’) on pavements. His explanation for this is that people refuse to “give up unilaterally their right of way until the last moment” as there is “a strong norm of bilateral accommodation in street behaviour”; in other words, people expect others to cooperate with them, and will not deal with the avoidance all by themselves: brushing into them may act as a reminder to the other that they are in the wrong, and should have moved in some way. In addition, as people often wait until the very last moment to avoid each other anyway it is inevitable that sometimes the procedure will go wrong, leading to a slight collision.

3.1.4 Avoiding people in front

A collision can occur with a person in front of the subject if the subject is walking faster than that person and approaching behind him. In these situations it is clear that the subject has two methods of avoiding a collision occurring: slow down, or keep walking and detour around the collidee. These two options will now be explored.

3.1.4.1 Slowing down

If the subject decides to slow down he will change his speed to be the same as the collidee, and will then walk behind him. However, Wolff noticed that people do not usually walk directly behind each other. Rather, the subject will "strive to maintain a head-over-the-shoulder relationship with the person in front", positioning himself so that his body is behind that person's shoulder rather than his head. In high density traffic it is sometimes not possible to move into this position, so in these circumstances people will angle their heads so that they can peer over the shoulder of the person in front, even if their bodies are near parallel. It is only if the subject is about 5 m behind the person in front that they will then walk directly behind him. An exception to this rule is if people are circumnavigating an obstruction or passing through a cluster of others, in which case people follow closely behind the lead person, in his 'wake', until the obstruction is passed. The two reasons for this head-over-the-shoulder behaviour given by Wolff are that it facilitates viewing what lies ahead, and it secures against stumbling into the feet of the person in front.

A feature that results from people slowing down is the formation of lanes, also described as clusters and pedestrian streaming, discussed by all three main sources. They consist of groups of people all moving in roughly the same direction at roughly the same speed. Older [Old68] noticed the "tendency for pedestrians travelling in the same direction to follow one another in files which interweave with those from the opposite direction, reducing the interaction between the flows." Collett and Marsh observed that "the walking speeds of individuals who constitute the main body of a lane will tend to be rather similar."

An example of lane formation is shown in figure 3.3 (reproduced from work by Still [Sti00]): lanes are shown with yellow arrows. It is clear that people have formed into lanes going in opposite directions, each lane only one person wide, with multiple lanes in each direction.

Collett and Marsh discuss the reason that lanes form by mentioning that the more crowded the pavement is, "the greater the advantage afforded to the individual who slots in behind those ahead of him." The idea is that it is much easier for a person to fall into line behind others than it is for him to make his own way through a crowd of people: no decisions about avoiding oncoming actors need to be taken, as long as he stays behind the person in front and behaves the same way he does. In the words of Collett and Marsh, while describing how a person has to monitor those around him: "the more well-defined the streaming [lane-formation], the more easily he will be able to ignore oncomers"; therefore,

Figure 3.3: Lane formation in a crowd; arrows indicate the direction of travel in a lane [Sti00]

"streaming may be seen as a corporate and informally constituted means of reducing the monitoring and locomotor problems of individual pedestrians." It is worth noting the phrase "informally constituted": people do not consciously decide to form lanes, they just emerge from the decisions made by all the people in them that it is easier to walk behind the person in front than it is to try to weave their way through a crowd at a faster pace.

Goffman notes that "traffic tends to sort itself out into two opposite-going sides", with the dividing line somewhere near the centre of the pavement, assuming there are equal flows in both directions. Wolff clarifies that the widths of the lanes are only equal if the number of people in both lanes is roughly equal: as the number of people in one of the lanes increases that lane "will occupy a greater width of the pavement than its numerical proportionality to the sparsely populated lane would predict."

Wolff, who thinks of lanes as clusters of people, observed that "clusters are not groups of related members, but are creations of various environmental contingencies." He notes that one way they can be formed is from a large group of people all waiting to cross a road; once they can proceed "the large cluster breaks into smaller clusters proceeding down the pavement at different speeds." Due to the difficulty of threading through densely packed clusters they "tend to grow by aggregation in the rear", as people approach the cluster, decide it is not worth trying to get round it, or through it, and slow down to match its speed. Finally, he states that in "periods of low density, there are fewer clusters than in periods of high density, and these clusters are composed of fewer individuals": this ties in with Collett and Marsh's view that lanes are more advantageous the more crowded the pavement is.

3.1.4.2 Overtaking

Sometimes, despite the clear advantages of forming a lane, a person will decide to overtake instead, the usual reason for this being a desire to walk faster than the current speed of the lane. Goffman believes that "those who want to pass almost always will have room to do so": even if there is not room to do so on the pavement itself, people can often make use of the road instead. However, there are some situations where it may in fact be impossible to overtake, such as in a narrow corridor.

Collett and Marsh describe the ways in which a person in a lane may overtake: either by weaving through his own lane, or by walking down the interface between opposing lanes, or by taking to the road.

As mentioned earlier, Wolff notes that people tend to walk behind the shoulder of the person in front; he also states that this pattern of people, even when compressed in a high-density situation, allows the maximum number of people to pass through a given area. It also allows them "the maximum freedom of movement for accommodating and for pursuing their individual goals." This freedom of movement allows a person "walking faster than a particular cluster to thread his way through the cluster, maintaining his own pace, but not forcing others to accommodate to him." However, if the cluster is too tightly packed then a "fast walker will either intrude upon the members of the cluster, forcing everyone in it to accommodate, or will intrude into the oncoming pedestrians in the opposing lane."

Finally, it is clear that in very low density situations a person will have a free choice whether or not to overtake the person in front of him, as long as he is confident that his overtaking manoeuvre will not cause a collision with an oncoming person. The decision to overtake will then be made solely on whether the person is willing to slow down or not.

3.1.5 Behaviour after the collision

After a collision has been successfully avoided, it might be expected that a person would simply continue walking in his new position on the pavement, possibly making minor adjustments so that he was still walking towards his goal. However, Wolff observed that the majority of people in his experiments attempted instead to revert to the original path (or 'line-of-walk') that they had been on before the avoidance behaviour started. Goffman also states that "when an individual momentarily shifts from a lane to facilitate traffic flow he is apparently likely to shift back into it after the interference is past."

Neither author is prepared to give a conclusive reason for this behaviour, and both are in fact slightly surprised by it: as Wolff states, "all paths on a pavement are of equal utility for the purposes of reaching the corner". However, both propose theories as to why it may occur. Wolff suggests it could be because people see the avoidance behaviour as a temporary adjustment to their behaviour, and their criteria for successful adjustment might include some notion of returning to the original behaviour: this seems to be a rather circular argument. He also proposes that people may have 'position preferences' with regard to "environmental boundaries and contingencies"; in other words, some people may prefer to walk as far from the edge of the road as possible, and will work to get back to that position after the collision has been avoided.

Goffman suggests another, rather more plausible, explanation: from his original path a person has already conducted "long-range scanning of approaching people; by returning to it, he can take up a preparedness already established." In other words, he has already worked out how to avoid a number of predicted collisions that would occur from continuing on his original path; if, on the other hand, he stayed on the new path he would have to re-establish these far-range predictions. Therefore, in the interests of minimising time spent thinking about avoidance behaviour, it is easiest to return to his original path.

Finally, Wolff observed one exception to these rules that occurs if a person had been about to turn off the pavement to, for example, enter a shop, when the collision avoidance took place. Following the rules above, one would expect the person to go back to the original path, walk the short distance to the point at which he would have turned off had he not been interrupted, and then turn towards his target. However, he will in fact walk straight towards the area of interest, ignoring his original path. This behaviour only occurs when the person is initially just a few metres away from the point at which he would turn off the pavement.

3.1.6 Evaluation

One of the problems with a lot of the research discussed above is that it consists largely of vague, qualitative statements; in addition, terms such as "pedestrian dance" are not defined, and there is sometimes disagreement among the authors: for example, Wolff states that both people *always* perform the step-and-slide, whereas Collett and Marsh contend that sometimes only one of the people will do so.

These issues hinder the task of converting these ideas into algorithms. However, by combining the descriptions of the various authors on each aspect of behaviour a general, and fairly consistent, picture of human behaviour emerges; this will be used in chapter 5 as the basis for the rules given to each actor.

3.2 Vision and attention

Much research has been performed on the ways in which the human eye works, and how information from the eye is processed by the brain. This section will start with a brief overview on the physiology of the eye, and then look at the higher-level control that sets where the eye should look.

3.2.1 The eye

An introduction to the human eye is given by Bruce and Green [BG90]. The eye consists of 120 million rods spread out over the whole retina, and 7 million cones. The rods have a high sensitivity, so can detect relatively small amounts of light, but not colour: only the cones can detect colour information. Given the limited number of cones, if they were spread out all across the eye their average distance apart would be relatively large, and so it would be hard to detect detailed colour patterns; instead most of the cones are concentrated in the centre of the retina, in a depression called the fovea, as shown in figure 3.4.

The principal function of the rods is to act as peripheral vision, covering the areas that are not currently visible in the fovea. The visual acuity of this peripheral vision is only 15-50% of the acuity of the fovea, according to Jacob [Jac95]. However, it is designed to be particularly reactive to flashing objects and sudden movements.

The fovea extends across a visual angle of approximately 2°: this is slightly less than the width of one's thumb held out at arm's length. It is therefore clear that detailed observations of an area require the eye to move so that the light from that region falls upon the fovea. This is known as 'foveating' areas of a scene, and is the reason for the familiar eye movements one sees as a person looks around a scene.

Bruce and Green also describe the different ways the eye can move. The eye is controlled by three sets of muscles, to allow horizontal, vertical and rotational movement. For completeness, all seven classified types of eye movements are given here, although only the first two will prove relevant:

- **Saccades**. These are the way the eye moves to a new part of a scene, and are sudden, rapid movements of both eyes at the same time. It takes 100–300 ms to initiate a saccade, and another 30–120 ms for the actual movement. According to Argyle and Cook [AC76], the speed of

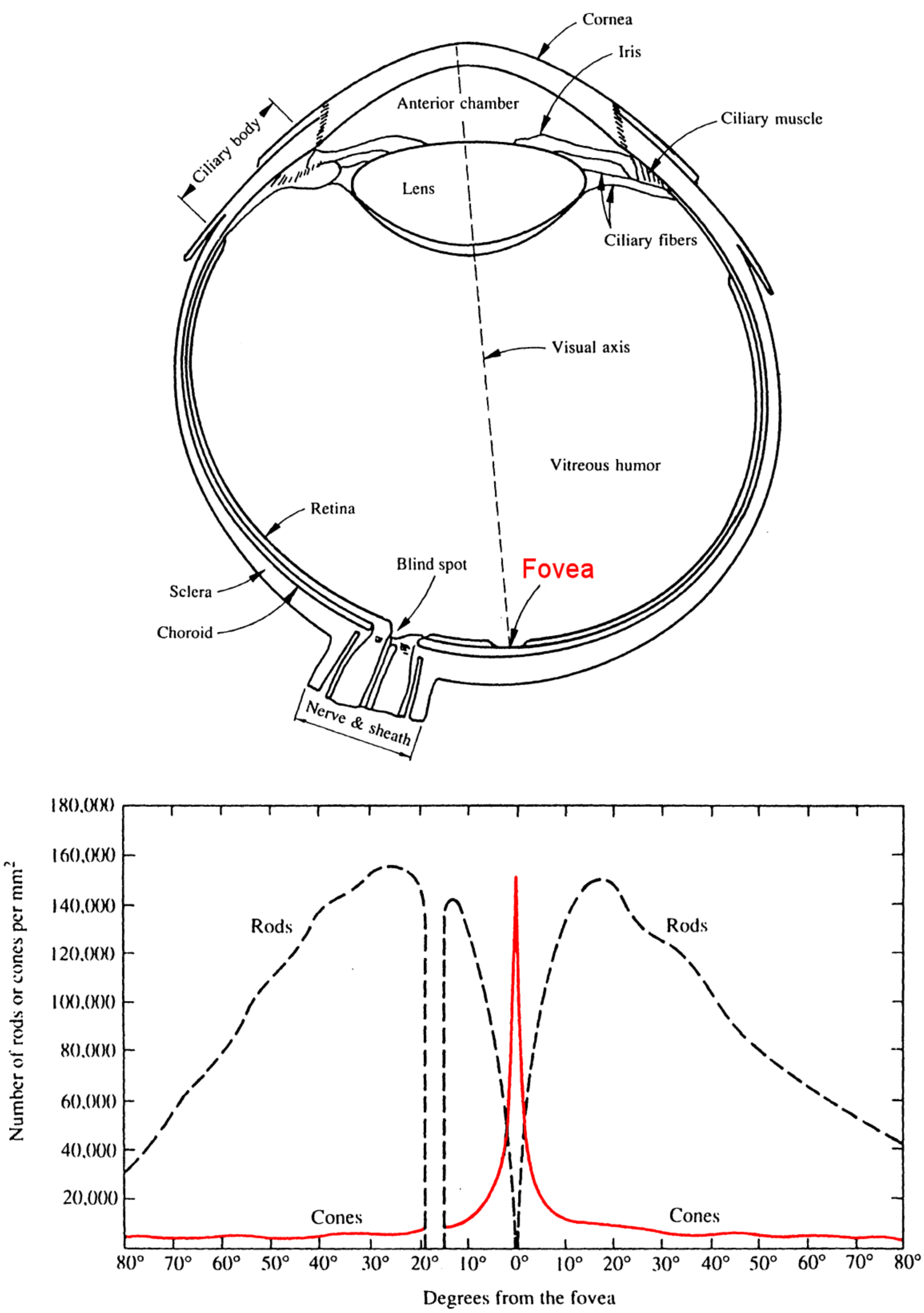

Figure 3.4: Top: The structure of the eye, showing the fovea. Bottom: the distribution of rods and cones across the eye [GW92]

rotation is up to 1000° per second, and the total shift is up to 20°. Once initiated they cannot be stopped, and the destination cannot be changed – which shows that the target must be fully chosen in advance. During the saccade, processing of the image is mostly suppressed, so the new location is actually observed after the saccade has finished, during a period of 'fixation', lasting for 200–600 ms.

- **Pursuit motion**. After a moving object has been fixated once using a saccade, the eye can perform smoother, slower (150° per second), lower latency movement to keep it foveated. Such movements cannot be induced voluntarily, but only triggered by a moving object. They allow the position of an object to be tracked for as long as desired.

- **Convergence**. This changes the distance from the eye that is currently in focus.

- **Nystagmus**. As the head turns, the eye performs smooth movement in the opposite direction to follow a position in the scene, followed by saccades to select new positions.

- **Rolling**. This is an involuntary movement to keep the field of view level as the angle of the neck is changed.

- **Drift and microsaccades**. Small amounts of drift may occur during fixations, and microsaccades bring the eye back to the current target.

- **Physiological nystagmus**. The eye undergoes continual high-frequency oscillations, shifting the image on the retina; if an image is fixed on the retina it will soon fade, so this ensures continual stimulation of the cones.

It should be noted that almost none of these movements is perceivable at a conscious level; one simply thinks of looking around a scene, and suitable movements are automatically generated.

3.2.2 Attention

Theeuwes [The93] explains that shifts of the direction of eye gaze are not the same thing as shifts of attention: a person's attention can be focused, at least for a short period, on a target other than the one currently being looked at. He suggests that there are two stages to the attention selection mechanism:

- a pre-attentive stage that operates across the entire visual field with no capacity limitation;

- an attentive stage with a limited capacity that can only deal with one item at a time.

In this mechanism the target for the next fixation is decided while the current target is still being examined. Attention will then switch to the target that has been decided upon, and only then will a saccade be triggered to actually look at it. This process is also described by Parasuraman [Par98]: once the attention selection process has chosen a target, that stimulus will be tagged; this tagging opens up a neural pathway, allowing access to awareness and memory, and letting that stimulus contribute to eye motor actions.

Koch and Ullman [KU85] also describe this two-stage process of selective visual attention: the eye first forms a set of topographical maps of the scene, known as an early representation, containing only information such as colour and disparity; in the second stage a particular location will be chosen to be fully attended, often based on the conspicuity of locations in the early representation.

3.2.2.1 Periphery capture

Kahneman [Kah73] discusses the ways that attention can be captured by items seen in the peripheral vision: this is known as periphery capture. There are two classes of stimuli that can attract attention: physical properties of an object and collative properties. Physical properties include factors such as size, colour and complexity of the object, and also motion: the eye is drawn more towards a moving object than a stationary one. Examples of collative properties are novelty, incongruity and significance. In particular, other people, particularly moving people, will capture attention; this is partly due to people being social animals and wanting to look at other people, but it is also useful, during walking, to determine if a collision will occur with other people. Yarbus [Yar67] observed that attention seems to be captured by "those parts of the scene that contain the most information for the perception of it": information that is "important for the recognition of the scene."

At this point it should be clarified that just because an object in the periphery has captured attention it does not mean that the eye will immediately move to look at that object. Egeth and Yantis [EY97] explain that it simply means that the attention selection mechanism now knows that the object exists, and may schedule an eye movement to look at it at some point in the future.

Kahneman notes that "a fixation is often determined on the basis of information previously acquired in peripheral vision" and cites an experiment carried out by Williams [Wil66]: subjects were shown a large array of figures, each including a number; the figures varied in colour, size and shape. On each trial the subject was told to look for a particular number, and was given some information about the figure containing the number, such as its colour. Subjects found it easy to restrict their fixations to figures of the designated colour, reducing the search time compared to not being given this information. This indicates that information collected in peripheral vision can be used to plan eye movements.

However, Egeth and Yantis confirm that, although some information on targets must be recorded for use in planning where to look, no accurate data about the position and velocity of the object is known at this stage. This will only be determined when the attention selection mechanism schedules an eye movement to foveate the target, and after the period of fixation has elapsed.

The research presented so far has been considering periphery capture in a general sense, not looking at any particular situation. However, if it is known that the situation in question is a person walking along a street, then one can more clearly define which areas of a scene will cause periphery capture. This was an area studied in detail by Goffman.

Goffman uses the word 'scanning' to define the way that people look at the scene around them while walking down a street. He proposes that a person walking down a street makes an assumption that those people "to the front of a close circle around him" are the ones whom he should look at, and that those "who are a person or two away or moving behind his sight-line can be tuned out." This scanning area extends "three or four sidewalk squares" in front of the person; this translates to between 4.5 m and 6 m. When somebody enters this scanning area he will be glanced at briefly, usually in a discrete fashion, and his position, direction and speed of movement determined. This check is done as early as possible, so that if collision avoidance does need to take place, it will only involve a small, and therefore undemeaning, change in direction of travel. If a person has been observed and a collision is not predicted then he can generally be ignored, unless both people are in a fast-changing situation where changes of direction are common. In particular, others can be safely ignored when they "have come close enough abreast of him so that any interference from them would require a very abrupt turn." In addition, a person "does not concern himself with oncomers who are separated from him by others": this means that in dense traffic people will ignore even those who are actually very close to them. As a result, Goffman concludes that the scanning area is in fact not a circle but an "elongated oval, narrow to either side of the individual and longest in front of him", with its area changing based on the current traffic density.

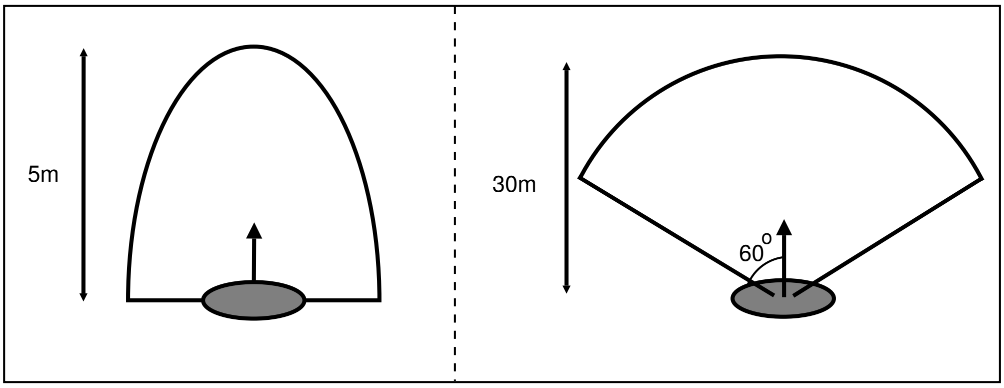

Figure 3.5: Areas scanned by actors; actors are shown as grey ellipses with a velocity arrow

The periphery capture oval described by Goffman is shown on the left of figure 3.5. It should not be confused with a person's peripheral vision: it is smaller than the region covered by peripheral vision, and it includes areas (to the side of the person) that would require some turning of the head to look at. Rather, it is the area that is deliberately scanned by a person to check for potential collisions.

Goffman does not mention how large the scanning area can grow in situations of low density. However, in the experiments carried out by Wolff described in section 3.1.3 it was found that people will start to avoid collisions when the potential collidee is up to 30 m away. It is clear that the collidee must have been observed before he can be avoided, otherwise he would not be known about. Therefore it follows that in very sparsely populated areas the scanning area may extend up to 30 m away from somebody. In these cases there is also no reason for it to be elliptical in shape, as everybody will be far away and not occluded by others; it will therefore be a sector of a circle, defined by the person's field of view. This is shown on the right of figure 3.5.

3.2.2.2 Types of looking

Kahneman describes four different types of 'looking' – high level control of eye movement – of which the first two are relevant for this work:

- **Spontaneous looking**

  This occurs when a person is viewing a scene without any specific task in mind. An object that has been seen in peripheral vision will be selected and fixated. The decision about which object to look at next is complicated, but seems to be based on how much information the person thinks he might be able to obtain by looking at it (rather than the actual importance of the object); this measure of importance seems to be much more of a factor in the decision than any physical feature of the object, such as brightness or complexity. Spontaneous looking is also guided by stored knowledge, for example that looking at other people will return useful information.

- **Task-relevant looking**

  This happens when a person has a specific task in mind when looking at a scene, and needs to find out particular information. While this is normally applied to searching for objects in static scenes, it also applies to a person walking down a street, who needs to check that he is not going to collide with anybody. In particular, once a collision has been predicted the person will keep looking back at the collidee to get up-to-date information about his new position and velocity; in a sense, monitoring that collidee has become his 'task'.

- **Orientation of thought looking**

  It has been observed that people will move their eyes based on what they are thinking about, with no relation to the scene in front of them.

- **Intentional manipulatory looking**

  People may use their eyes to point at an object to indicate it to somebody else, in a similar manner to a hand gesture.

3.2.2.3 Monitoring

As mentioned in passing above, once another person has been looked at once, his position and velocity will be known; however, both of these will change over time, particularly during collision avoidance manoeuvres. Therefore it is necessary for all observed people to be monitored until it is certain that they will not move on to a collision course (or until it is confirmed that current avoidance techniques will be sufficient).

This is a topic explored by Kahneman. He first considers the related task of a pilot faced with a panel full of dials, all of which need to be monitored. It turns out that a trained pilot's gaze is distributed among the dials so that the probability of looking at a particular dial will be proportional to the time since it was last looked at, the importance of the information presented on that dial, and the rate of production of that information; in fact, Senders [Sen65] showed that pilots closely approximated a theoretically optimal monitoring policy. In the more general case, Kahneman sums up as follows: "When fixation is governed by a visual task, the locus of fixation is determined by an assessment of the probabilities that relevant information will be acquired, and that the acquired information will be useful."

Kahneman lists five factors that are used to assess the likelihood that looking at a particular area will yield useful information:

- the elapsed time since the area was last scanned;
- the detection of some features of a possible target;
- a preliminary identification of a target;
- known base rates for a particular area (i.e. how likely it is that there will be new information);
- general knowledge about the structure of the situation.

Kahneman also gives an example: suppose an adult is reading a newspaper while sitting beside a swimming pool, and a baby is randomly crawling about nearby. At what intervals will the adult look up to check the baby's position? He states that the interval "will depend on where the baby was last seen, and on its direction and rate of progress at the time." Other factors are also important, such as the depth of the pool, and whether there is any water in it.

Research by Yarbus revealed similar effects: he noticed that eye movement occurred in cycles, with the most important parts of the scene being scanned first, and then re-scanned continually, rather than time being spent fixating the unimportant parts. The route that the eyes take while performing this scanning is known as a 'scanpath'. His research showed that a large proportion of viewing time was spent following this scanpath, with brief diversions to look at new areas, which might then get incorporated into the scanpath. He found that the time spent on one cycle could be up to several tens of seconds. This would seem to be the same effect observed by Kahneman, just described slightly differently; it seems clear from both that people spend most of their time monitoring targets that they have already seen, with some time devoted to looking at new targets from peripheral vision.

3.2.2.4 Evaluation

At this point it is worth briefly reflecting upon the evidence outlined so far. The model of vision presented is one in which only objects being looked at by the fovea can be studied; objects in peripheral vision can sometimes (although not

Figure 3.6: Simons and Chabris' gorilla experiment [SC99]

always) capture attention, but no detail about them can be discerned until they are foveated. This is perhaps surprising to many: common sense suggests that people have continual, detailed vision across their entire field of view. However, a number of experiments have proved that this is not the case.

Yarbus conducted an experiment in which a test subject was asked to watch an object in the centre of his field of view; objects in peripheral vision were then changed. In the majority of cases, these changes were not observed by the subject. Grimes [Gri96] found that observers often do not notice when two people in a photograph exchange heads, as long as they are not being specifically watched at the time; even though the change may catch the eye, the subject is unaware of what has changed. In a very extreme case, Simons and Chabris [SC99] asked subjects to watch a video of six people passing a basketball and count how many passes took place; at a point during the video, a woman in a gorilla suit walked across the playing area, sometimes edited to look partly transparent. In the transparent cases (left frame of figure 3.6), only 8% of the subjects noticed the gorilla. When it was left opaque and stopped to beat its chest (centre and right frames of figure 3.6), still only 50% noticed it. This effect, known as inattentional blindness, has been confirmed in similar experiments by many other researchers.

However unlikely it sounds, the model presented does indeed seem to be a valid representation of human vision and attention.

3.3 Moods and emotions

Research by psychologists on the effects of moods and emotions on human behaviour proved more elusive than the topics in the preceding sections. This may be due to a belief that the effects are 'obvious' – 'everybody knows' what being afraid means. However, this hypothesis suggests an alternative approach to discovering what these effects are: interviewing a selection of people to determine if there are common shared beliefs about behaviour.

This section first presents the limited findings from psychology research, and then gives details of a set of such interviews carried out by the author.

3.3.1 Psychology literature

Emotions Ekman [Ekm99] studied emotions for many years. He defines the purpose of emotions as being to inform others of "what is occurring inside the person (plans, memories, physiological change), what most likely occurred to bring about that expression (antecedents), and what is most likely to occur next (immediate consequences, regulatory attempts, coping)." This is the same as the purpose of simulating emotions: to allow viewers to get insights into the mental state of the actors.

Ekman also describes the concept of emotion families, each made up of a set of related states. The same approach is used by Baron-Cohen et al. [BCGWH04]: they set out 412 emotion concepts, divided into 24 categories, each group containing related emotions. (This categorisation was carried out to create a DVD demonstrating emotional states to people with autism.) The 24 emotion categories are: Afraid, Angry, Bored, Bothered, Disbelieving, Disgusted, Excited, Fond, Happy, Hurt, Interested, Kind, Liked, Romantic, Sad, Sneaky, Sorry, Sure, Surprised, Thinking, Touched, Unfriendly, Unsure and Wanting.

Emotion characteristics It proved almost impossible to find any psychology sources giving explanations of what effect emotions have upon behaviour. The only source was a study carried out by Shaver et al. [SSKO01]. They hold the opinion that the best source of information about emotions is asking people for their opinions: they note that "people around the world can reliably name the emotion being expressed" in a photograph, indicating that there are shared beliefs about what constitutes an emotion. However, they caution that "everybody knows what an emotion is, until asked to give a definition" (quoted from work by Fehr and Russell [FR84]). Therefore, rather than asking direct questions they asked 120 subjects to write accounts of emotional experiences, either personal or generic, within their five selected emotion categories of: Afraid, Sad, Angry, Joyful and Loving.

Features of the specified emotion that were mentioned by a majority of the test subjects were:

- Afraid: "Eyes darting, looking quickly around", "Fleeing, running, walking hurriedly", "Feeling nervous, jittery, jumpy".

- Sad: "Lethargic, listless", "Slow, shuffling movements", "Slumped, drooping posture", "Withdrawing from social contact".
- Joyful (Happy): "Being courteous, friendly towards others", "Doing nice things for other people", "Being talkative, talking a lot", "Physically energetic".

The emotions Angry and Loving are not considered in this work, for reasons discussed in section 3.3.2.3.

3.3.2 Interviews

As a result of the lack of research on the effects of different emotions, the decision was made to conduct some informal research into this area.

3.3.2.1 Interview form

The interview consisted of two parts:

1. Participants were asked which emotions they thought they would be able to detect just by looking at somebody they saw on the street. The aim of this question was to see how many of the 14 emotions selected for simulation (see section 7.1) would be mentioned, thereby confirming their importance.
2. For each of the 14 emotions participants were asked how they would know if somebody they were watching was feeling that emotion, concentrating particularly on walking behaviour and eye-gaze movement. Emotions were presented in pairs to allow participants (if they chose to do so) to contrast their opposite effects on behaviour, rather than having to list all the effects in both positive and negative forms for opposing emotions. Emotion pairs were presented one at a time, and in a random order, so that participants would not be influenced by forthcoming emotions. Participants were allowed to talk until they had run out of ideas.

In total 10 people were interviewed – a mixture of students and older people, with a roughly equal balance of men and women; none of them were involved in this type of research. The original plan had been to conduct more interviews, but it soon became clear that everybody was mentioning the same ideas. This was a pleasing result, as it confirmed the theory that there are common ideas, shared by everybody, as to what emotions mean, and how they are manifested. It was therefore decided that there was no need to spend time interviewing any more people, as any extra insights would probably have been minimal.

3.3.2.2 Results

The emotions listed in the first part are shown below along with frequency counts. Emotions are listed using the standardised words used by Baron-Cohen, and some very similar emotions were combined: Afraid includes Anxious and Nervous, Unsure includes Confused and Hesitant, Carefree includes Relaxed, and Unfriendly includes Impatient.

Emotion	Count
Happy	10
Afraid	9
Angry	6
Sad	5
In love	5
Hurrying	4
Unsure	4
Carefree	2
Unfriendly	2
Drunk	1

For the second part, the following characteristics were given for each emotion, presented here along with a frequency count. Any ideas mentioned by only one person have been omitted. Very similar features are grouped together.

Afraid	Walking quickly	9
	Eyes darting around	9
	Looking behind	8
	No eye contact Quick furtive glances	5
	Looking further away	2
Carefree	Walking slowly	5
	Looking straight ahead Not looking round much	5
Thinking	Looking straight ahead	9
	More likely to collide with others	6
	Walking slowly	5
	No eye contact	2
	Not concentrating on walking	2

Excited	Looking around	9
	Hurrying	5
	Bouncy walking	3
Bored	Eyes fixed ahead	8
	Looking down	7
	Walking slowly	6
	Not noticing others	5
Interested	Looking around	10
	Paying more attention to others Looking at others	5
Sad	Looking down	8
	No eye contact	4
	Not paying attention to others	2
	Deliberately avoiding people	2
	Not stopping to talk to people	2
Happy	Looking around	5
	Making eye contact	4
	Acknowledging other people	4
Strolling	Looking around	8
	Walking slowly	7
	Stopping to talk to people	5
	Slowing down rather than overtaking	4
	Leaving room around others	2
	Making eye contact	2
Hurrying	Walking quickly	10
	Pushing through crowd Darting through gaps	8
	Walking boldly	5
	Not acknowledging others	4
Unfriendly	Barging past others Expecting others to move out of your way Bumping into others Pushing way through crowd	9
	No eye contact	7
	Not acknowledging others	2

Kind	Stopping to let others pass Negotiating how to avoid collision	7
	Making eye contact Looking at others	5
	Leaving room around others	2
Unsure	Looking all around	10
	Walking slowly	3
	No eye contact	2
Sure	Taking risks Walking boldly	8
	Walking straight towards goal	7
	Looking only straight ahead	5
	Walking quickly	5

3.3.2.3 Discussion

A number of interesting findings came out of the interviews:

- When asked to suggest emotions in the first part, participants mentioned 7 out of the 14 emotions selected for simulation (see section 7.1); only three other emotions were given: Angry, In love, and Drunk.
  - Although Angry was suggested by over half of the participants it was not selected for simulation, as all of the effects which it would have on behaviour would already be covered by the Unfriendly emotion – pushing through a crowd, not making eye contact, and not acknowledging others.
  - In love was also commonly mentioned. However, the effects of this emotion upon behaviour would only be visible for people walking in couples, a situation not covered by this simulation.
  - Drunk was only mentioned by one participant. Given this, and uncertainty about whether it qualifies as an emotion, it was not included.

  These results indicate that the 14 selected emotions encompass nearly all those which people would expect to see, as well as containing less obvious ones; in other words, the chosen emotions appear to be sufficient for the purposes of this simulation.

- In the second part, the degree to which each participant listed the same characteristics was surprisingly high: at least 8 of the participants mentioned the same feature for 11 out of the 14 emotions. This implies that there are indeed shared beliefs about emotions. Not only did this make the task of interviewing easier, by cutting down the number of people required to get useful results, but it also makes the task of simulation possible: if people had different opinions on the manifestation of emotions, it would be impossible to know what behaviours to give to the virtual actors. As it is, the chances of people being able to identify emotions on virtual actors look promising.

- As well as being useful as descriptions of the various moods, the responses from the second part also proved useful in suggesting which aspects of the behaviour of an actor should be variable – aspects both of eye movements and of walking behaviour. These behaviour variations will be explained in full in section 7.3.

The results of the literature survey and interviews described in this section were used to construct a table linking emotions with variations in behaviour; this is described in section 7.2.2.

3.4 Summary

This chapter started with an investigation of the ways people avoid collisions while walking, discussing detouring, the step-and-slide manoeuvre, overtaking and slowing down, and studying the behaviour of people after a collision has been successfully avoided. Some of the effects caused by avoidance behaviour, in particular lane formation, were also mentioned.

There then followed a section describing some of the physiology of the human eye. The key piece of information is that the eye cannot attend to its entire field of view at the same time; instead, it successively foveates areas of interest. The way in which the attention of the eye is brought to these new areas was then discussed, looking at the way objects appearing in peripheral vision are noticed and may then be looked at some time afterwards. The techniques used to monitor a scene were described, showing the ways in which people choose how often known targets should be looked at so that information about them may be kept up to date.

Finally, there was a brief look at the study of moods and emotions: the limited amount of research by psychologists, and the results of interviews carried out to obtain more information.

Chapter 4

Vision and Attention

The overall aim of this work is to produce a realistic simulation of human behaviour. Most aspects of behaviour are based on knowledge of one's surroundings, and so one of the elements required to simulate behaviour is an accurate representation of human perception. For the majority of people the most important sense for determining the content of their surroundings is vision, and so visual perception will be the principal focus of this work. Some information also comes from audition, such as hearing footsteps approaching from behind, but this less important sense will not be considered here (see section 8.2.1).

Visual perception can be considered as a two-level process:

1. Vision is the low-level process concerning the actual movements of the eyes.

2. Attention is the high-level process by which a person chooses where in his field of view to look so that he can obtain useful information.

In this chapter the ways in which both of these processes have been simulated are described, with reference to chapter 3 to explain the psychological basis for all decisions made.

4.1 Overview

An overview of the vision and attention system is shown in figure 4.1. The entire vision and attention system is simulated as a set of communicating agents; each actor has a set of these agents, and agents for separate actors are completely independent.

- The **Periphery capture** attention request agent analyses the exact positions of other actors held by the simulation, determines which are in the

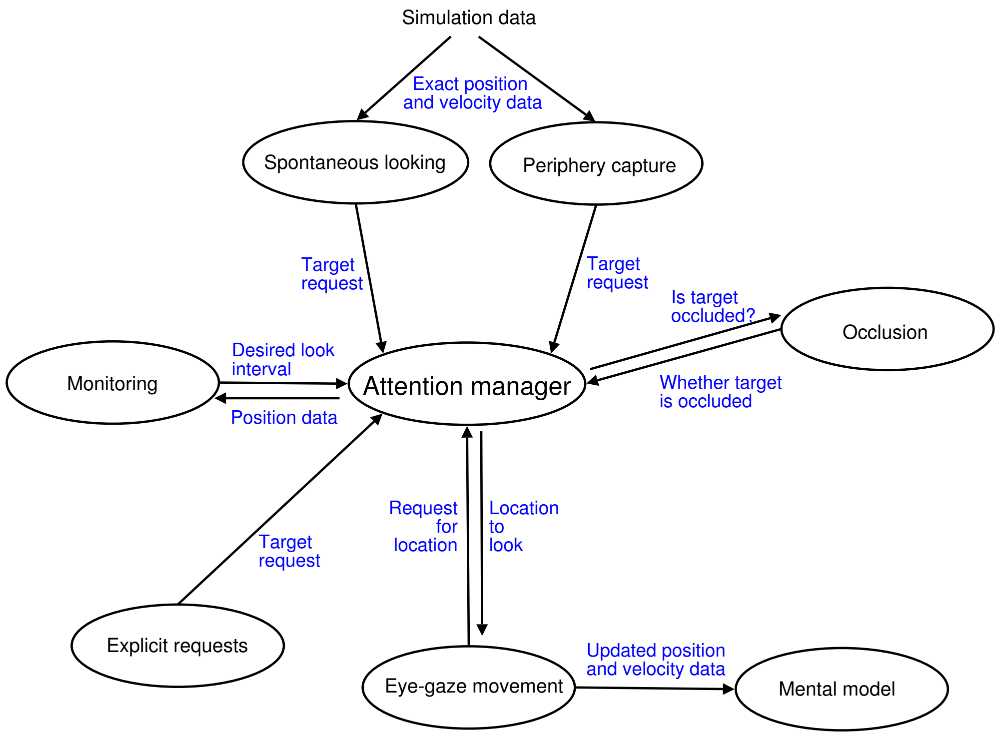

Figure 4.1: Attention agent architecture overview

viewing ellipse defined by Goffman (section 3.2.2.1), and sends requests to the attention manager for these to be looked at.

- The **Spontaneous looking** attention request agent is similar, but makes requests for all actors in the field of view to be observed.
- The **Monitoring** attention request agent sends re-observation requests to the attention manager, containing its desired interval between looks, based on position information sent to it by the attention manager.
- The **Explicit requests** attention request agent allows the actor to send a request to the attention manager for a particular target to be looked at.
- The **Eye-gaze movement** agent asks the attention manager where the eye should look next, and receives back the next target.
- The **Occlusion** agent receives a potential target from the attention manager, and returns whether that target is occluded or not.
- The **Mental model** is sent updated position and velocity information by the eye-gaze movement agent, and also performs its own interpolation of existing data.

- The **Attention manager** receives requests from the attention request agents, and determines which request has the highest priority when asked for the next target by the eye-gaze movement agent.

4.2 Vision

The low-level processes of vision are handled by the eye-gaze movement agent. Before describing this in detail, it is worth considering why vision needs to be simulated at all.

The movements of the human eye were described in section 3.2.1. The key point to be noted is that the eye, consisting of a small highly sensitive fovea and less sensitive peripheral vision, can only study objects (or people) that are currently being foveated – that is, light from them is falling onto the fovea, because the eye has been moved to look in their direction. Targets for the human eye to look at are selected from information gathered in peripheral vision, but this only provides an approximation of the targets' positions, not exact position and velocity data.

This work re-creates these limitations of the human visual system; this means that the eyes of the actors must continually move to learn about their surroundings.

For real people shifts in the direction of gaze will often consist not only of eye movements but also of turning the head, allowing the eyes to look in a greater variety of directions. In this system the actual movements of the eyes are not visible, and are instead shown by a purple line above the actor, pointing in the direction of gaze: this will be assumed to include any necessary head movements.

4.2.1 Eye-gaze movement

The eye-gaze movement agent is responsible for controlling the actual movements of an actor's eyes; it is passed details of where to look next by the attention manager (section 4.3.2).

The attention manager sends the eye-gaze movement agent a vector representing the direction in which to look. This is compared with the current gaze direction to calculate the angle through which the gaze must shift: this information is needed to work out how long the gaze shift will take.

As described in section 3.2.1 the main way in which the eye moves to look at a new area of interest is using saccades – sudden, rapid movements taking between 100 ms and 400 ms to complete. The amount of time taken is proportional to the angular distance travelled.

To reproduce the fundamentals of real saccades, in this system it is assumed that the eye can move through an angle of 50° in each frame of the simulation. Given a frame rate of 20 frames per second, and therefore 50 ms per frame, this gives a speed of 1000° per second, the figure given by Argyle and Cook [AC76] as the speed of saccades. Although this does not reproduce all the subtleties of saccades, such as the time taken to initiate them, or the associated head movements, it captures the essence of eye movements taking a non-zero amount of time, and the time taken is approximately correct.

Therefore the gaze movement time is calculated as:

$$\left\lceil \frac{\text{gaze shift angle}}{50^\circ} \right\rceil \text{frames} \qquad (4.1)$$

This rounds the time up to the nearest integer so that even short movements will take at least one frame.

Once a real saccade has been started, it cannot be stopped. In this system the result of this real-life limitation is that even if a high priority request comes in it will not be attended to until after the current eye movement is completed. Therefore, while a saccade is being carried out the job of the eye-gaze movement agent is simply to move the eye through a further 50° each frame.

The reason for actually moving the eye gaze direction each frame, rather than simply jumping to the final position after the calculated gaze movement time, is so that the movement of gaze will be visible in the graphical simulation.

When the eye gaze movement is finished the actor will be looking at the requested target. However, it cannot immediately acquire data about the target: first it must undergo a period of fixation. During the saccade movements of a human eye most visual processing is disabled so that the eye does not have to deal with very fast moving images. Once the movement is finished the fixation period lasts between 200 ms and 600 ms. For this system a delay of four frames (200 ms) is introduced during which time the eye remains static.

After the fixation is over the actor can be assumed to know about its target; in particular it will now know its position and velocity, and these are added to the mental model each actor keeps of its surroundings (see section 4.3.4). It is worth considering whether the actor should be able to discern the velocity of its target after just one look at it; this is a topic about which psychologists seem uncertain. However, it seems reasonable to take the view that the target will have moved sufficiently during the fixation period that a rough estimate, at least, can be made of its velocity, which may then get refined after further observations. In this system it is assumed that the actor will be able to acquire the target's velocity after the first observation, in basic accordance with this theory.

There is one exception to the condition that the target must be fixated, which occurs if a target is being tracked. This situation arises when the actor has a particular interest in the target, such as when it is trying to avoid a collision with it. In these conditions the human eye would use pursuit motion (described in section 3.2.1), in which the eye performs smooth, slow motion to keep a particular target foveated. During pursuit motion the eye can process what it is seeing, so it will continually get up-to-date information. This effect is re-created by setting the gaze shift time and fixation time to zero if the gaze target is the same as the previous target, and the actor is performing monitoring; this means that the actor will receive up-to-date information on the target each frame.

Each time the actor looks at a target it will record the latest values of position and velocity, updating the possibly incorrect values predicted by its mental model (section 4.3.4). The new information will also be used to update how often the target should be monitored (section 4.3.1.3).

Usually an actor will continually be given targets to look at, even if they are just random glances. In the rare event that the eye-gaze movement agent is not passed any target to look at by the attention manager, the default behaviour is simply for the actor to look straight ahead, in the same direction in which it is walking.

4.3 Attention

The psychology behind human attention was discussed in section 3.2.2, in particular the ways that a person's attention is drawn to different parts of his surroundings; the agents that reproduce this behaviour are now described.

The agents themselves are omniscient, in that they know the positions of all other actors in the world; however their task is to carefully filter this information so that they only pass on to the actor the information of which the actor should be aware, to simulate the way that real people do not have full knowledge of their surroundings. This limitation was made clear from the psychology literature discussed in section 3.2.2; the advantages of this approach will be discussed in section 4.3.5.

4.3.1 Attention requests

The function of the attention request agents is to collect external data from the world, or internal data from an actor's behaviour controller, and submit requests to the attention manager: each request represents a desire to cause a shift in eye movement to look at a particular part of the world which that agent believes to be interesting; in a sense the agents are competing for attention. There are four such agents, the first two concerned with external data, the last two with internal data.

- The *Periphery capture* agent inspects a small, key area of the actor's peripheral vision for actors that have not yet been observed.
- The *Spontaneous looking* agent examines the rest of the actor's field of view for other actors that may be looked at if there are no higher priority attention requests; this allows the actor to look around 'randomly'.
- The *Monitoring* agent repeatedly makes requests for actors that have been observed once to be observed again, to allow their new positions and velocities to be found. It must decide which actors need to be looked at more frequently than others.
- The *Explicit requests* agents are used by the behaviour controller to request an observation of a given actor, or to look in a particular direction; the results of these requests are used to decide upon appropriate behaviour.

Each agent is now described in detail.

4.3.1.1 Periphery capture

The psychology behind the idea of periphery capture was discussed in section 3.2.2.1. The concept is that, although only objects currently being foveated by the eye can be observed in detail, objects in the eye's peripheral vision can draw the eye's attention. Any object may draw attention, but Kahneman [Kah73] noted that it is primarily 'interesting' objects that do so: of particular interest are other people, especially moving people. Evidence collected by Kahneman, and other experiments described by Egeth and Yantis [EY97], showed in addition that attention is not always drawn immediately, but rather that the selection mechanism now knows that the object exists, and may subsequently choose to look at it.

These ideas can be translated neatly into an agent system in which an agent monitors all actors that move into the peripheral vision of its subject actor and sends requests to an attention manager for each of them to be observed when convenient: attention is not shifted immediately, but the information is present when needed.

Section 3.2.2.1 also described work by Goffman, which defines the area within which people may be subject to periphery capture: it is an elongated oval, or ellipse, centred on the subject, with its major axis along the current direction of travel, extending between 4.5 m and 6 m in front, and ignoring the half of the ellipse behind him (shown in the left of figure 3.5).

Bearing the above in mind, the mechanism of the periphery capture agent will now be described.

In each frame of the simulation the periphery capture agent considers all the actors in the scene. Any actors behind the subject are immediately filtered out, as are any more than 6 m away (the maximum reach of the peripheral vision ellipse). The purpose of the filtering is so that the check on whether actors are inside the ellipse only needs to be done on a small number of actors; this improves efficiency, as the ellipse check is more computationally expensive than the very simple angle and distance tests.

For this system, the periphery capture ellipse has a semimajor axis length of 5 m, and a semiminor axis length of 2 m. To work out if another actor is within this ellipse, it is necessary to transform the position of that actor into the frame of reference of the subject, in which the local x-axis points down the major axis of the ellipse.

The procedure for this calculation is shown in figure 4.2: the top half of the diagram shows the absolute positions of the two actors; the lower half shows the position of the target actor relative to the subject, with both rotated so that the semimajor axis of the subject's peripheral vision ellipse (shown here in grey) is aligned with the x-axis.

- The angle of the subject's velocity with the x-axis (β) is calculated in advance, and used for all the tests with other actors.
- The angle of the vector from subject to target with the x-axis (α) is found for each target, based on their relative position (x and y).
- The distance between the subject and target (d) is found.
- The position of the target in the subject's frame of reference (x' and y') is found from d and $(\alpha - \beta)$. ($(\alpha - \beta)$ is used rather than $(\beta - \alpha)$ to give an angle in the correct direction.)

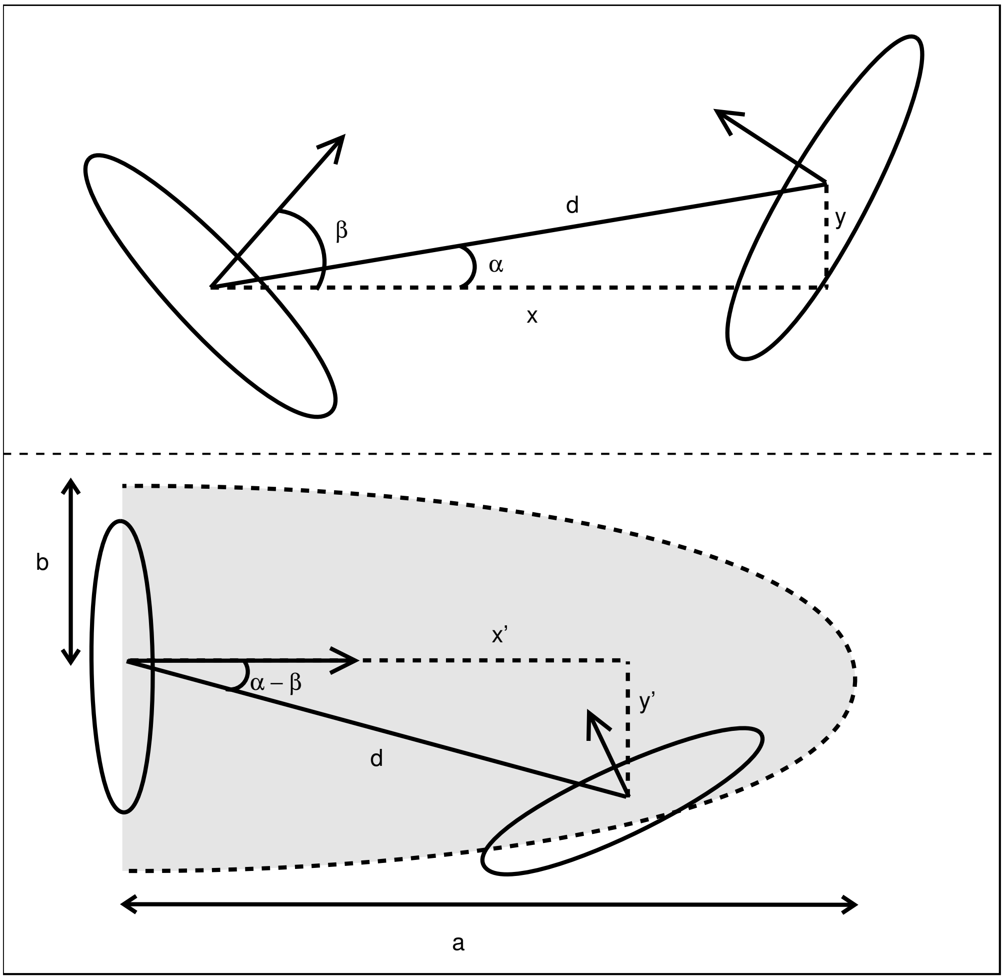

Figure 4.2: Determining if an actor (right) is within the subject's (left) peripheral vision

Checking whether the target is within the subject's viewing ellipse is now simply a matter of substituting x' and y' into the equation of an ellipse:

$$b^2x^2 + a^2y^2 - a^2b^2 = 0 \tag{4.2}$$

where a is the semimajor axis length, and b is the semiminor axis length. Therefore the resulting equation is:

$$b^2x'^2 + a^2y'^2 - a^2b^2 < 0 \tag{4.3}$$

which, if true, means that the target is inside the ellipse – as is the case in figure 4.2.

At this stage the periphery capture agent checks to see if the target has been detected before – if it is either already in the queue of actors found in peripheral vision waiting to be looked at, or if it is already being monitored by the actor. If not, a request is sent to the attention manager to add it to the periphery queue, containing the unique *id* of the target actor.

In cases where the attention manager is already aware of the actor its 'age' is set back to zero, to show that it is still in the actor's field of view. This will be explained further in section 4.3.4.

In addition to recording the *id*, a 'rough' position of the actor is also stored. It has already been stated that the accurate positions of actors cannot be determined until they are foveated. However, it is clear that some rough information must be stored, so that the eye will know where to look when it does choose to observe the target; observations (such as those by Yarbus [Yar67]) have shown that the eye does not need to search for targets by using a large number of small movements, but will perform a saccade to look straight at the target. In this system the actual position of the target actor is recorded by the periphery capture agent, and used later to control the eye movement to observe it. However, this position data is *not* used in any other way by the system, and so will have no effect upon behaviour; only the position found when the target is actually foveated is used to control behaviour.

A new rough position is also recorded when the attention manager is already aware of the actor, so that the eye will always know where it should look to perform an observation.

4.3.1.2 Spontaneous looking

In section 3.2.2.1 it was noted that under certain conditions people that are outside the viewing ellipse defined by Goffman may be observed. Evidence was presented from work by Wolff [Wol73] that collision avoidance may start when a person is up to 30 m away; it is clear that somebody has to be observed before they can be avoided. From personal experience it is obvious that people look further ahead than the 6 m maximum given by Goffman. Rather than thinking of Goffman's viewing ellipse as being the *only* area wherein observations are made, the correct interpretation is to see this as the highest priority area for observations to be made. In other words, if there is a potential target in the ellipse it will be looked at rather than a target outside the ellipse; but if there are no targets in the ellipse then more distant targets may be observed.

To implement this behaviour the attention manager needs to know which actors are present in the subject's field of view, so that it has the option of observing them; therefore the spontaneous looking agent should record details of these actors.

After the periphery capture agent has processed all of the actors, the spontaneous looking agent studies those that are not contained within the viewing ellipse (the majority). Two checks are made to see if they are within the actor's field of view:

- The distance between subject and target actors must be less than 30 m.
- The angle between the direction of travel of the subject and the vector between subject and target must be less than 60°.

The shape of the field of view is shown on the right of figure 3.5. Unlike the periphery capture area it is a sector of a circle, not an ellipse. This is because it is far less constrained, and is simply providing a resource of actors that could possibly be looked at, rather than actors that should be observed with any degree of urgency.

If an actor is found to be in the field of view, and is not currently in either the periphery queue or the spontaneous looking queue, a request is sent to the attention manager to add it to the spontaneous looking queue; this request contains the unique *id* of that actor, and its rough position. As with the periphery capture agent, if the attention manager is already aware of the actor then its age is set to zero and its rough position is updated.

4.3.1.3 Monitoring

The human activity of monitoring was described in section 3.2.2.3; its purpose is to continue to observe other people after they have been looked at once, so that any changes in velocity can be noted. This is important, as a change in velocity will lead to a different course, and the possibility of a collision that would not have been originally predicted. In particular, during collision avoidance both people involved will be continually changing their velocities, so frequent observations are critical to stop a collision from happening.

The factors used to determine the frequency at which other actors should be monitored in this system are based on those listed by Kahneman [Kah73]:

- time since last observation;
- predicted time until collision with subject.

The first of these factors is one given by Kahneman; the second is based on his theory that the frequency of monitoring will be dependent on the likelihood of gaining new information, and on knowledge about the situation. In this case, the situation is collision avoidance, and the knowledge in question is that the closer the other actor is the more useful information about it will be. The weightings of these factors will be described in section 4.3.2.2.

The monitoring agent is responsible for keeping track of when all actors were last looked at, and storing information about when they should be observed again. It is called by the eye-gaze movement agent (section 4.2.1) after an actor is first looked at (either from periphery capture or spontaneous looking). It sends a request to the attention manager to add the actor to the monitoring queue; this request contains:

- the unique *id* of the actor;
- the amount of time before another observation is needed.

The attention manager will in addition store the time at which another observation should be made, by simply adding the current time to the interval specified.

The decision about the length of the interval is based upon the distance between the subject and the target: clearly the further away the target is the less often it needs to be observed, as there is less threat of a collision. The interval ranges from 1 to 30 frames, with the maximum value being given to any actor further than 5 m away. (If a collision has been predicted then this calculated interval will become less important, so if a far-away actor is on a collision course it will still be looked at frequently; see section 4.3.2.2 for details.)

The monitoring agent is also called each time an actor has been monitored. This enables the interval until the next observation to be updated based on the new distance between subject and target, giving more frequent looks as the actor gets closer.

4.3.1.4 Explicit requests

In section 3.2.2.2 various types of looking were described, as different forms of high-level control of the eye. Spontaneous looking, where a person is viewing a scene without a particular task in mind, has already been covered in section 4.3.1.2. Some aspects of task-relevant looking have also been handled: periphery capture is task-relevant, as the task in question is avoiding people while walking; another aspect of that task is to monitor people to see if a change in their course will cause a collision. However there is one more aspect of task-relevant looking: shifts of gaze which are under the control of the person. This can either be conscious decisions to look at somebody or, more often, a subconscious choice needed to make a decision about future behaviour.

For example, a conscious gaze shift could be to look at a friend who has just been noticed, to look at a traffic light signal, or to look into a shop window. Subconscious gaze shifts can be used to check if people are walking in a group, or to check if there is room to overtake somebody: these are both used by the behaviour controller, see sections 5.2.1 and 5.2.4.2 for more details.

To re-create this behaviour it is therefore necessary for the behaviour controller to be able to send explicit requests for gaze shifts to the attention manager. These are divided into two types: actor requests and direction requests.

Actor requests If the subject wishes to make a gaze shift to look at one particular actor then the explicit request agent is passed the unique *id* of that actor, along with a priority value; the agent stores this information in a queue. The priority is used by the attention manager (section 4.3.2) to choose between different explicit requests, but both types of requests always have priority over other types of gaze shift. The priority is simply a number between 0 and 100, with 100 being the most urgent, and it is up to the actor's behaviour controller to choose the value.

Direction requests Sometimes an actor will want to look at a particular target without knowing if there is in fact an actor to look at (or at least without knowing details of that actor). In the above example of checking to see if it is possible to overtake somebody, the actor does not know if there is anybody to look at – that is the reason for the check. Even if the actors it is checking for have been detected by the periphery capture or spontaneous looking agents they will not yet have been observed, so no details will be known about their positions and velocities.

All the actor knows in advance is the direction in which it wants to look. Therefore the actor passes the explicit request agent a vector representing that direction, along with a priority, as above. The attention manager will look at any actors close to this direction vector – for details see section 4.3.2.2 [Explicit requests].

4.3.2 Attention manager

The attention manager is in overall control of all the attention agents; its functions are in two main areas: receiving requests from the attention request agents, and choosing where the next gaze shift should be directed.

4.3.2.1 Receiving requests

The attention manager can receive requests from all four attention request agents discussed in section 4.3.1: periphery capture, spontaneous looking, monitoring and explicit requests. Each type of request is stored in a different queue (with explicit actor requests and explicit direction requests separated); this makes it easier for the attention manager to consider only requests of the type it is interested in.

The periphery and spontaneous-looking queues simply store the unique *id*s of all the requested targets. The explicit actor request queue in addition stores the priority, and the explicit direction request queue stores the direction and priority. The monitor queue stores more information on when the actor should next be looked at.

4.3.2.2 Next target

The most important function of the attention manager is to return the next target to look at, following a request from the eye-gaze movement agent. The decision is based on results in psychology; all of an actor's choices will be linked to the behaviour of real people.

To look for the highest priority target, the five queues are examined in the following order:

1. Explicit actor requests *or* Explicit direction requests, based on priority (see below)

2. Periphery capture *or* Monitoring, based on their relative scores (see below)

3. Spontaneous looking

It should be clear that requests that have come directly from the actor should be given priority over all others, as they are triggered by the behaviour controller requiring more information before it can decide what course of action to take. It is also clear that spontaneous looking should only be done if there are no other necessary vision tasks – by definition, this is the activity that the eye performs when it has nothing else to do. The most important target is found from each of the periphery capture and monitoring queues; the one with the higher score (described below) is chosen.

The procedures for choosing the highest priority target in each category are now discussed.

Explicit requests To decide which explicit request to deal with first, the attention manager simply compares all of the stored priority values in both queues (actor and direction), and picks the request with the highest value; contention is dealt with by choosing the request that has been waiting longest.

If an explicit actor request has the highest priority then this actor will be returned as the next target to look at. Only those actors in the explicit actor request queue which are currently in front of the subject are considered: under normal circumstances actors do not look behind themselves (although see section 7.3.1.4 for exceptions to this rule).

If a direction request is found to have the highest priority (and as long as that direction is still in front of the target) then the next task is to find out which actor in that direction to look at. As explained previously, the idea of the request is that the subject is looking in this direction to check for actors that it does not already know about, to make sure that its model of the scene is correct. For a real person, this would simply be a case of looking in the given direction and then inspecting any people whom he had not seen before.

Due to the constraints of the system, the subject cannot simply look in the given direction – it has to have a specific actor to look at. Therefore, to re-create the effect, a search is made through all actors that have not yet been observed by the subject, and that are within 15° either side of the specified direction vector. (This figure was chosen to ensure that any actors in the rough direction given will be found, rather than the direction having to be set precisely.) From these actors, the one that is nearest to the subject is chosen: this is because the nearest actor is the one most likely to have an effect upon the subject's immediate behaviour, and therefore the one that would be inspected first by a person.

An explicit direction request will only lead to one unobserved actor being looked at; therefore, if the behaviour controller wishes to thoroughly check a particular direction for actors it must send more than one request to the attention manager. There is no danger in sending more than necessary, as if no actors are left to be found in that direction the request will be ignored and another gaze target will be selected from another queue. By only looking at one target, more control is given to the behaviour controller, which should be in a better position to make decisions.

An example of an explicit direction request is shown in figure 4.3. The direction vector is shown with a thick black line, with two vectors 15° either side of it to show the search area. In this example, it is assumed that the green actor has already been observed, so the choice is between the red and blue

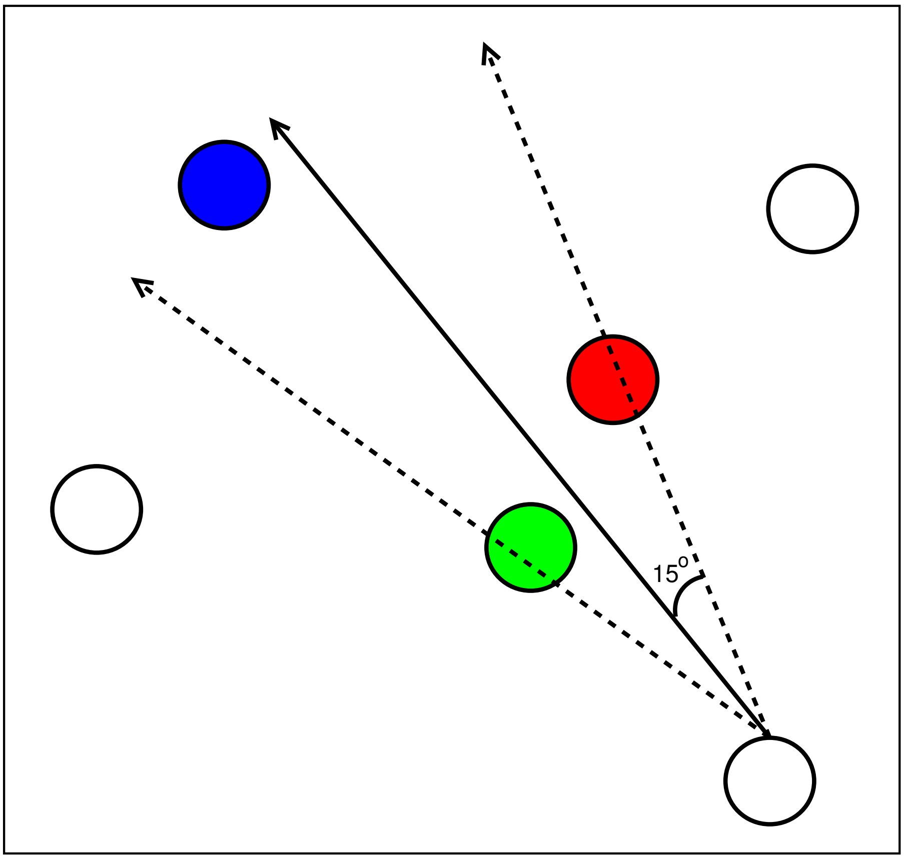

Figure 4.3: Choosing which actor to look at following an explicit direction request

actors; the red is nearest to the subject, so it is chosen.

Periphery capture The process by which a person chooses which of the targets from his peripheral vision to look at first is unclear. What is known is that people do have a choice about which target to look at next, as explained in section 3.2.2.1; it also appears that the most 'important' target will be chosen first [Yar67]. As the targets have not been inspected yet, no accurate details of position or velocity are known; the only available information is a rough idea of size, based on the angle of the retina subtended by the target. It seems clear that people would choose to look at the target that appears largest, as this is likely to be the most important.

To simulate this effect, the attention manager looks through all the targets in the periphery capture queue, calculates the distance between each of them and the subject, and chooses the nearest one; this is the same as choosing the target that appears largest, as the largest target will also be the closest (because people are all roughly the same size).

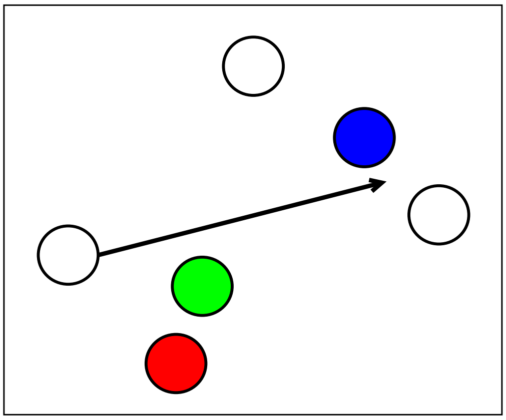

Figure 4.4: Selection of actors for periphery capture and spontaneous looking

In figure 4.4, the subject's gaze direction is shown with an arrow, and the green actor has already been observed. In this case the red actor would be chosen as the nearest.

Clearly an actor that is further away could be travelling much faster than the nearest actor, and therefore be more important to look at to determine if a collision will occur, but no details about velocity are yet known, so this cannot be taken into account. The only information to base the decision upon is the rough position that was passed by the periphery capture agent, representing the limited amount of information people detect in their peripheral vision; therefore a rough distance is all that can be worked out.

After the nearest actor has been found, it is given a score so that it can be compared with the most urgent monitoring target. The score is calculated from the distance (capped to a maximum of 10 m) as:

$$\text{score} = \frac{10.0\,\text{m} - \text{distance}}{10.0\,\text{m}} \times 4.0 \tag{4.4}$$

so that the score varies from 0.0 (if the target is further than 10 m away) to 4.0 (if the target is right on top of the subject). Actors further than 10 m away are still considered interesting, but they are far enough away that their importance will not vary based on distance.

Explanation of the reasoning behind the weighting of this and following equations will be given following the presentation of the remaining equations.

Monitoring To choose the target that should be monitored next, the attention manager makes use of the information stored in the monitoring queue for each potential target: the predicted time to collision, and the time at which another observation was requested by the monitoring agent (see section 4.3.1.3). For each actor the following procedure is followed:

- Check that the actor is in front of the subject, else ignore it.
- If a collision is predicted between the subject and the actor, record the predicted time to collision.
- Calculate how long has passed since the next observation of the actor should have occurred.
- If this time is negative *and* no collision has been predicted this actor can be ignored, as there is no need to look at it yet.
- Calculate a score based on the time to collision, ranging from 0.0 for a time of 50 frames (or over) to 2.0 for a (theoretical) collision in zero frames. (Beyond 50 frames, all collisions are considered to be far enough away to be of equal importance.)
- Calculate a score based on how overdue the observation is, varying between 0.0 for an observation that is not overdue and 1.0 for one overdue by 20 frames (or more). (Beyond 20 frames, all observations are considered to be so overdue as to be of equal importance.)

The actor with the largest sum of these two scores:

$$\text{score} = \left(\frac{50.0 - \text{time to collision}}{50.0} \times 2.0\right) + \left(\frac{\text{time overdue}}{20.0}\right) \tag{4.5}$$

is chosen as most urgent to be looked at. These two scores try to balance the conflicting demands of looking at actors which are overdue to be observed (who could have moved onto a collision course since their last observation), while also looking at those actors for which a collision is already predicted. The different scaling factors applied to calculate these scores ensure that an actor with which a collision is imminent will always be given priority over an actor that is merely overdue for an observation (as shown below).

Periphery capture v. Monitoring Once the best targets have been found from the periphery capture and monitoring queues the scores calculated for each of them are compared, and the larger is selected as the gaze target. If neither returned a result then the actor will perform spontaneous looking; if only one returned a result, that target is selected.

Spontaneous looking The *Next target* function only gets to this stage if no target has been selected from any of the previous categories: no requests have been made, there are no new actors in the viewing ellipse, no collisions are predicted, and no actors are overdue to be observed again.

In these situations, the attention manager looks through all the actors in the spontaneous looking queue. For all actors that are in front of the subject the angle between the current gaze direction of the subject and the vector between the subject and the actor is calculated. The smallest of these angles represents the smallest change in gaze direction required to look at the target, and therefore the most appropriate target. For the moment it will be assumed that this is the desired behaviour, as the subject is more likely to notice a target that is near to its current gaze direction. However, this is altered by later work on moods and emotions (see section 7.3.1.1).

In figure 4.4 the blue actor would be chosen as needing the smallest angle change from the current gaze direction.

Equation construction All of the equations use simple linear proportionality, for two reasons:

- This is the simplest and most obvious approach; there is no evidence to suggest that people use any more complicated techniques.
- Experiments (detailed below) were carried out, observing which targets the subject would look at when given a range of different types of requests. To work successfully it should look at important targets frequently, but not ignore any targets completely; this was found to happen.

The weightings also needed to be carefully designed. The weighting on the *time overdue* score in equation 4.5 was arbitrarily set to a unit scale. *Time To Collision* was then set to be twice as important: this ensures that targets with imminent collisions are looked at increasingly often, but not to the absolute exclusion of other overdue targets; this was confirmed experimentally (see below).

The maximum periphery capture score in equation 4.4 was set to 4.0, to be just larger than the maximum monitoring score, 3.0; this is designed so that an actor suddenly appearing very close to the subject and in its viewing ellipse will be looked at with more urgency than any monitoring request, as it will be unknown whether the actor is on a collision course: if it is, a collision may be imminent. These values, being close together, are designed to ensure the most urgent periphery requests get priority, without always giving periphery targets

Number	Target	Type	TTC	Overdue
1	22	Periphery		
2	9	Spontaneous		
3	7	Spontaneous		
4	7	Monitoring	25	-26
5	7	Monitoring	27	-26
6	**49**	**Periphery**		
7	7	Monitoring	33	-26
8	**22**	**Monitoring**		**14**
9	7	Monitoring	35	-26
10	7	Monitoring	37	-26
11	**34**	**Request**		
12	7	Monitoring	40	-26
13	7	Monitoring	42	-26
14	**22**	**Monitoring**		**6**
15	**9**	**Monitoring**		**5**
16	7	Monitoring	46	-26
17	7	Monitoring	49	-26
18	67	Spontaneous		

Figure 4.5: The eye movements of one actor during the course of a simulation

higher priority than possibly more urgent monitoring requests; once again, this was confirmed experimentally.

Experimental testing To confirm that the equations described above were performing correctly, the targets which a particular actor was looking at during the course of a simulation were studied. An extract from the data obtained is shown in figure 4.5: 'Number' is a count added for reference purposes, 'Target' is the *id* of the actor being looked at, 'Type' shows the type of observation, 'TTC' shows the Time To Collision with the chosen target (if relevant), and 'Overdue' shows how overdue the chosen target was for a monitoring observation (with a negative value showing how long until it would become overdue).

The subject starts with no knowledge of the scene, so its first priority is to look at the one actor (22) within its periphery capture ellipse; following this, it performs spontaneous looking, during which it observes actors 9 and 7. A collision is predicted with actor 7, and so it monitors this actor.

- Observation 6 demonstrates that actors from periphery capture can be more important than monitoring known actors.
- Observation 8 shows that actors that are overdue for re-observation can become more important than actors for which collisions are predicted –

in this case actor 22's overdue value has grown sufficiently high that it takes priority.

- Observation 11 shows that explicit requests always take priority.
- The Time To Collision for actor 7 gradually increases until, in observations 14 and 15, even actors with fairly small overdue values take priority.
- Finally, at observation 18, a collision with actor 7 is no longer predicted, and the subject goes back to spontaneous looking.

These results (along with many other runs of the simulation) confirmed that the weightings were correct: the subject would mainly monitor its collidee, but it would also consider actors found through periphery capture, and actors overdue for re-observation.

The full set of results from which the above table was produced is given in appendix B.5, along with a longer run.

These equations are clearly ad hoc in design, rather than being based on psychology: it would be impractical to determine this kind of detail on how people choose which target to look at next. However, they are based firmly on experimental data, and match known aspects of people's attention choices.

Occlusion As the attention manager is checking for targets in all of the queues, it has to consider whether the chosen actor is actually visible to the subject. Therefore in each of the attention categories, once the best actor has been chosen it is sent to the occlusion agent (section 4.3.3), to be checked for occlusion by obstacles and by other actors. If it is found to be occluded then the next best target will be searched for within the same queue or, if no other suitable target can be found, in lower priority queues. Results of the occlusion check are cached during the search, so that they do not need to be repeated within the same iteration.

Output When the best, unoccluded, target has been found, the direction vector between it and the subject is found, representing the desired gaze direction. This vector, along with an indication of the type of gaze, is passed to the eye-gaze movement agent. The type of gaze can be important, as monitoring gaze shifts may use simulated pursuit movements, rather than saccades, as described in section 4.2.1. If the actor was from any queue other than the monitoring queue it is removed from that queue.

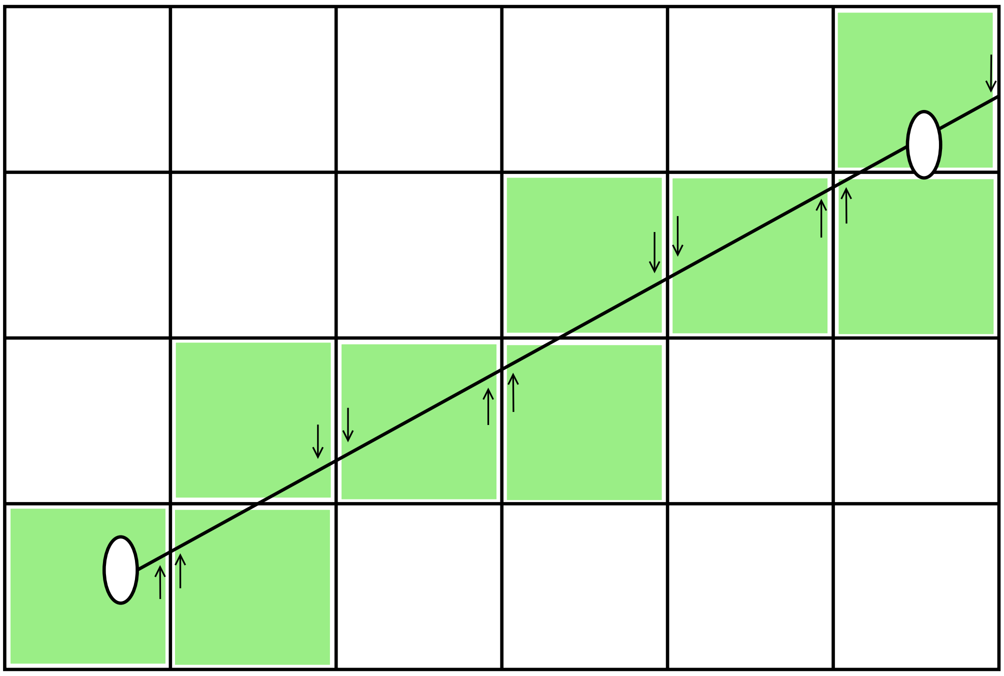

Figure 4.6: Process of checking for occlusion by obstacles

4.3.3 Occlusion

The occlusion agent is used by the attention manager to determine if the target actor will be visible to the subject actor, or if the line of sight is blocked by obstacles or other actors, in which case a new target will need to be selected.

4.3.3.1 Occlusion by obstacles

The layout of the world in which the actors exist is defined by a grid of squares, and so obstacles (such as walls) consist of one or more grid squares (see section 5.4.2.2 for an explanation of the world map). To check if there is an obstacle occluding the view of the target it is simply necessary to see if the line between the two actors passes through any square containing an obstacle.

To work out which grid squares the line of sight passes through it is first necessary to determine which octant the line lies in; for the purposes of this example it will be assumed it is in the first octant.

In this case, the point of intersection of the line of sight with the right-hand side of the map square containing the actor is found. Points on the line just to the left and right of this point are found – shown in figure 4.6 by arrows. The squares containing these points (shaded in green) are checked to see if they contain an obstacle. The process is then repeated by adding the width of a square to the x-coordinate of the intersection point, and ($width\ of\ a\ square\ \times$

Figure 4.7: Some parts of the person standing behind are visible, highlighted in red

gradient of line) to the y-coordinate, and checking the squares either side of this point. This continues until the square containing the target actor is reached.

If any square along the line contains an obstacle, then the line of sight to the target actor is blocked, the procedure is abandoned, and a new target must be selected. This method is robust: even if the line of sight only passes through a small amount of an obstacle square, it will be detected; this is shown in figure 4.6: all the squares containing the line are marked.

4.3.3.2 Occlusion by actors

The target actor can also be hidden from view by other actors that are positioned in front of it. However, the situation is more complicated than simply checking to see if the line of sight intersects any other actors. Firstly, even if the centre of the target actor is hidden, if the actors are not perfectly aligned then other parts of its body will be visible. Secondly, even if the target is standing directly behind another actor, it will probably still be visible through gaps in the occluding actor, as shown in figure 4.7. To take these two factors into account it was decided to check all the potential occluding actors to see if any of them are *completely* blocking the target actor; if *two or more* of them are, then the target will not be visible by the subject. If just one actor is in the way, it is assumed that some part of the target, such as an arm, leg, or part of the head, will be visible. This technique was chosen as it is a much cheaper operation than a full visibility determination that would consider the shape of the actors; it is also quicker than the false-colour rendering process described in section 2.2.3.5. Although it is less accurate than these approaches, the saving in time outweighs the small loss of accuracy.

It is not necessary to thoroughly check all of the actors in the scene to see if they occlude the target: most will be nowhere near it. Therefore a first filtering stage removes all those actors not in the subject's field of view – principally those actors behind the subject – using a simple angle check. Then all actors which are further away than the target are removed.

Any actors remaining after this filtering then undergo tests to see if any of them completely block the target. For each potential occluder a check is done to see if any part of the target can be seen around it. For the purposes of this check each actor is treated as a circle: this makes the algorithm much simpler than dealing with any more complicated shape, and should have very little effect on the results. The process works as follows (with reference to the examples shown in figure 4.8):

- Find the vectors between the centre of the subject actor and the occluder, and between the subject and the target (black lines).

- Work out whether the vector to the occluder is to the left or right of the vector to the target; in both examples, it is to the right.

- If the occluder is to the right, find the vectors between the centre of the subject actor and the left-hand side of the occluder, and between the subject and the left-hand side of the target (blue lines). If the occluder is to the left, the right-hand side is used.

- Check these new vectors to see if they are the opposite way round to the original vectors; in the examples, where the occluder is to the right of the target, the test is to see if the new vector to the occluder is now to the left of the new vector to the target.

- If this check fails, as in the top half of the figure, some part of the target will be visible; if it succeeds, the target is completely occluded by the tested occluder, as in the lower half of the figure.

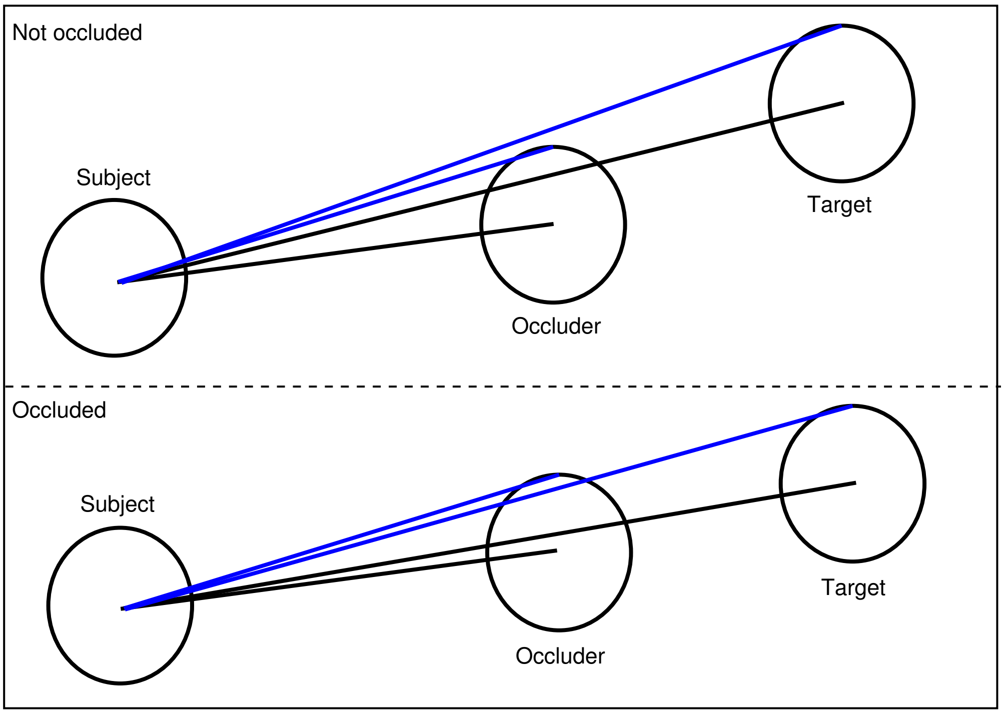

Figure 4.8: Process of checking for occlusion by actors

If it is found that the target is blocked by *two* occluders no more checks are made, and the agent reports that the target is not visible.

(A subtlety in the calculation of the angles of vectors is mentioned in appendix A.3.)

4.3.4 Mental model

In the real world each person has to build up his own picture of his surroundings by continually inspecting the scene with eye movements. The process of building the picture is subconscious: one simply *knows* where people are. However, it is clear that the only source of information on where other people are comes from what has been seen with the eyes. (For the purposes of this discussion the small contribution of the sense of hearing will be ignored, as, in normal conditions, it is far less significant than that from the eyes; see section 8.2.1.) In other words, there is no global source of information: each person has to work out for himself where other people are, and where they are headed. The implications of this will be discussed in section 4.3.5.

To re-create this effect it is necessary to give each actor its own model of the scene, which it creates for itself. This is in contrast to the simplistic approach of allowing each actor to access the database of actor positions and velocities

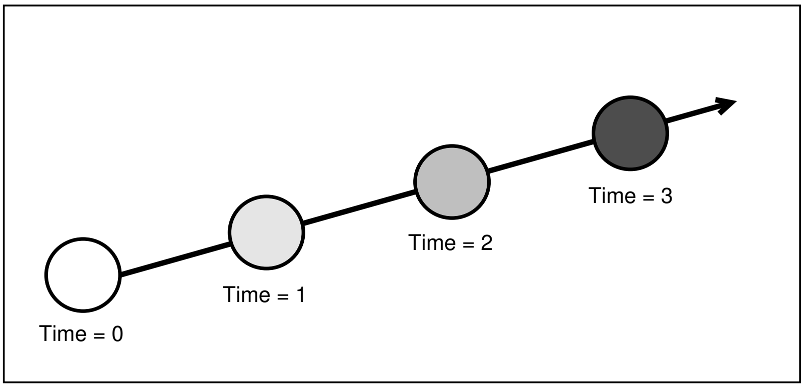

Figure 4.9: Interpolation of mental model positions

kept by the simulation. The model is kept up to date in two ways: by updates, and by interpolation.

Updates When an actor has undergone fixation by the eye-gaze movement agent (section 4.2.1), the observed details are passed to the mental model. In particular, the current position and velocity of the actor are observed, and replace the values previously held in the mental model.

Interpolation On each frame of the simulation at most one actor will be observed; frequently no actors will be observed if the eye is partway through a saccade or performing fixation. Yet it is still necessary to attempt to keep the mental model up to date, so that appropriate behaviour choices can be made. Therefore each frame the mental model updates the positions of all actors (other than any that have been observed that frame) based on the last recorded velocity value.

In figure 4.9 the position of an actor is shown as it would be predicted by the mental model over a period of frames; the velocity stays constant, and is added to the last known position.

It seems certain that a similar procedure must be performed by people to keep their model of their surroundings up to date; as has already been made clear, it is only possible for people to look at a very small area of the scene at a time, and everything else must be based on predictions. These predictions, of paths that people will take, and future collisions, can be simulated by a mental model in which the position and velocity of each actor is always known, either definitely as the result of observations, or approximately due to interpolations.

Common sense would suggest that an actor should be able to compare the predicted mental model positions with information from peripheral vision; any significant differences would then cause a shift in gaze direction to investigate. However, there is no evidence in any of the psychology literature discussed in section 3.2 to support this view: position and velocity data can only be determined when a person is being foveated; peripheral vision cannot provide this type of information. Therefore in this system the mental model position of an actor will only be updated when it is monitored in due course by the subject.

Data The mental model holds a variety of information about each actor:

- Position and velocity: observed velocity, and observed or predicted position.
- Rough position: when an actor has been detected in peripheral vision by either the periphery capture or spontaneous looking agents its position is recorded here; this position is not used to affect behaviour, just to assist the attention agents.
- Age: this records the number of frames since the actor has been present in the subject's peripheral vision (see below).
- Active: whether the actor has been seen in peripheral vision at all. This is set to true when the actor is first seen by either of the peripheral vision agents.
- Observed: whether the actor has been fixated by the eye. This is set to true by the eye-gaze movement agent.
- Time To Collision: if a collision is predicted between the subject and this actor then the number of frames until this will happen is stored.
- Collision position: where the predicted collision would happen.
- Being avoided: whether attempts are being made to avoid the actor.

The fields specific to collision avoidance will be discussed in chapter 5.

Age If an actor is seen by the periphery capture or spontaneous looking agents then its age recorded in the mental model is set to zero; this indicates that the actor is currently present in its peripheral vision. However, during the interpolation of new information that is carried out on the mental model each frame, the age is incremented by one. Therefore, if an actor has moved out of the subject's field of vision then its age will slowly grow, showing how long it is since it would have last been visible. It should be noted that this is not the

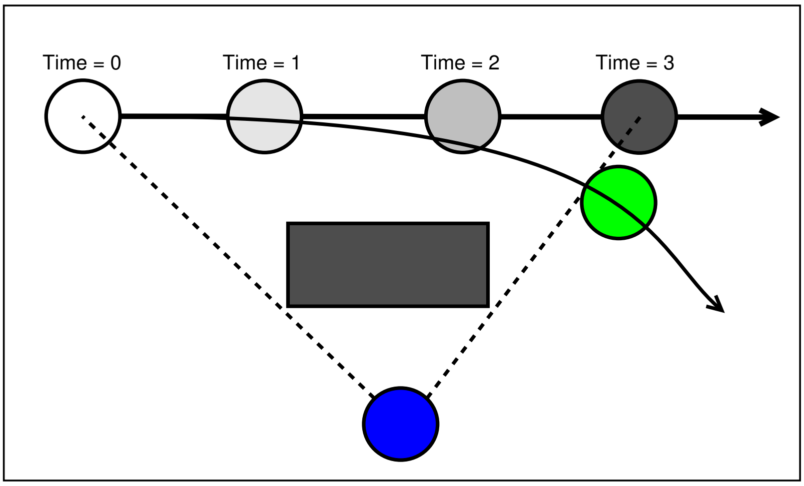

Figure 4.10: 'Incorrect' interpolation of positions while occluded

time since the actor last *was* looked at, but since it *could have* last been looked at.

Once the age reaches a threshold value, currently set at 20 frames, the actor is removed from the mental model, and a request is sent to the attention manager to remove it from any queues (including the monitoring queue) it may be in. This is done so that the subject will not keep on trying unsuccessfully to look at an actor that is no longer visible: it would not be able to see it anyway, so it should not waste effort trying.

The reason for allowing the age to grow to 20 frames before removing the actor is to simulate the real effect whereby people can infer the position of another without them actually being visible. For example, if somebody has passed behind a wall, a pillar, or somebody else, for just a brief period of time, a person would assume he is continuing at the same pace and therefore be able to predict at what point he would reappear on the other side of the obstruction. However, over time this prediction of the other's position will grow increasingly uncertain, due to either an inaccurate initial estimate of velocity, or due to the velocity changing since the last observation; after a certain amount of time the prediction will be effectively useless. In this system, this is assumed to happen after 20 frames, a fairly low figure. However, it should be noted that as soon as the actor becomes visible again, for example by reappearing from behind a wall, it will be seen again by one of the peripheral vision agents, and will therefore soon be added back to the mental model.

Figure 4.10 shows an example of the incorrect prediction of the position of an actor (by the blue actor) as it passes behind a wall. Based on its velocity at time = 0, it predicts that it has kept going in a straight line while behind the

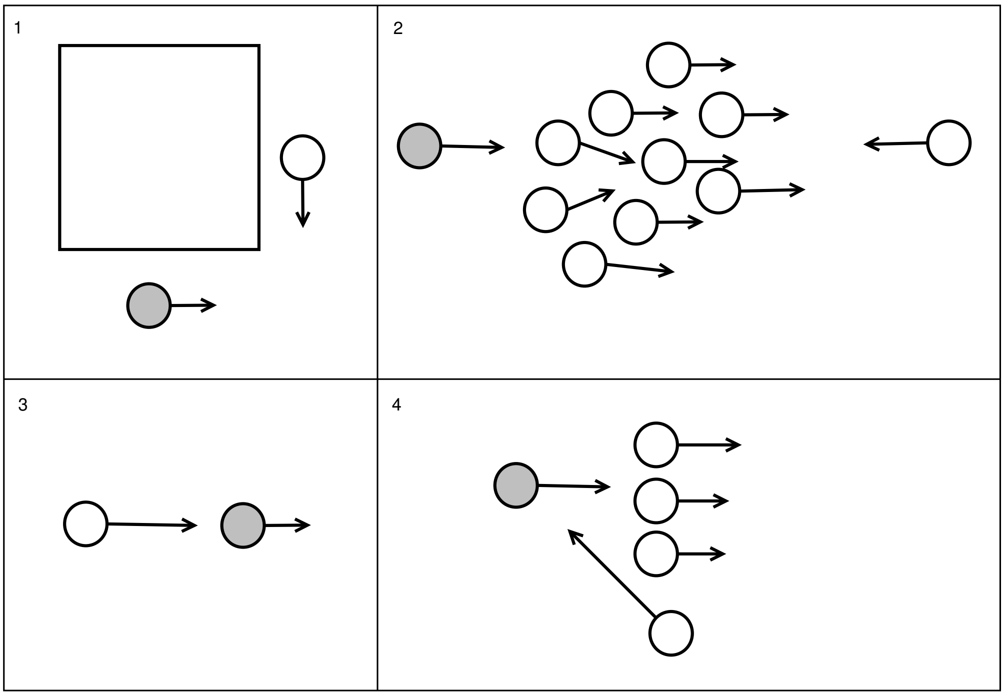

Figure 4.11: Four situations demonstrating advantages and disadvantages of different vision methods

wall – shown by the grey positions; in fact, it has changed velocity, and emerges at the position shown in green. This example shows how quickly predictions can become inaccurate, and hence why the age is only allowed to grow to 20 frames; up until this point, the prediction is still useful for giving a rough idea of position – here, the actor is still relatively close to its predicted position.

4.3.5 Discussion

The simulation of attention, and the use of the mental model, has a number of advantages and disadvantages, which are now discussed.

4.3.5.1 Comparison to other methods

The clear advantage of this system is that it is more closely based on real human behaviour than other systems. In the following discussion of different techniques that have been used in the past, four situations will be referred to, illustrated in figure 4.11:

1. Two actors approaching the corner of a building from adjacent sides should not be able to see each other, as their view of each other is blocked by the wall. Therefore they should not alter their courses in an attempt

to avoid the collision: they should either actually collide, or be forced to swerve out of the way at the last minute.

2. An actor that is part of a crowd of others all moving in the same direction should not be able to observe a far-off actor moving in the opposite direction, as the view of that actor will be blocked by the other actors in the crowd.

3. If an actor is being approached from behind by a fast-moving actor it should not be able to see it, and should therefore not alter its course.

4. In the real world, a person may be concentrating so hard on a particular situation (such as a collision, or a person of interest) that he does not notice another person with whom he might collide, even though that person is in front of him and not occluded by anything. Actors should re-create this effect, only noticing the collision at a late juncture, and therefore having to take more drastic avoidance action than normal, or actually colliding. In the example, the subject is concentrating on the three actors moving away from it, and does not notice the actor approaching from the side.

Total awareness Each actor in a total awareness simulation is allowed to access the accurate position and velocity information held in the model. It is able to 'see' actors behind it and actors behind walls, and instantly know as soon as an actor changes direction. This is the simplest approach, as all the information needed is already stored in the system, but it is also the least realistic; it was the approach used by Reynolds [Rey87] in his original Boids model. The main problem is that actors possess information that they could not possibly have in real life. In example 1, if the actors have total awareness of others they will, unrealistically, avoid each other. In example 2, the far-off actor will be visible at all times, and avoidance behaviour will start much too early. In example 3, an actor with total awareness will incorrectly avoid the actor behind it. Finally, with total awareness it is impossible to simulate example 4.

In summary, actors react in situations where, in the real world, they would either react later or not at all.

Field of view An improvement upon total awareness is only allowing the actor to receive information about the actors which are in its field of view, either defined simply as being in front of the actor, or within a 120° arc – 60° either side of the actor's direction of movement. This will prevent incorrect behaviour in example 3, above, but does nothing to solve the problem of occlusion. An example of this approach is Reynolds' later work on steering behaviours [Rey99].

Simulation of occlusion Some systems will base the information given to actors on the complete information held in the model, but restrict how much the actor is able to see based on principles of occlusion, both with obstacles such as walls, and with other people. This will prevent incorrect behaviour in examples 1 and 2 and, in conjunction with a field of view simulation, example 3 as well. This is the approach taken by the majority of researchers, although the methods of determining occlusion vary. One approach is analysing the database of actor positions to determine which actors are currently visible: this was used by Lamarche and Donikian [LD04] in a technique involving Delaunay triangulation and ray casting. The alternative approach is to render the scene and use computer vision techniques to determine occlusion (see section 2.2.3.5) – the approach used by Massive [Mas03].

Mental model None of the above techniques is able to simulate correct behaviour in example 4, as this relies on not just knowing what is *visible* to the actor, but also what it has actually *seen*. This can only be done using the simulation of attention and the mental model approach described here. For examples of situations in which the use of this approach has an effect on behaviour, see section 6.2.1.2.

4.3.5.2 Apparent limitations

With the advantages of the mental model come a number of apparent drawbacks; these will now be examined, and the reason why none of them is in fact disadvantageous will be explained.

The first thing to note is that the data held by an actor in its mental model will usually be different from the actual positions and velocities of the other actors – it will be incorrect. The general reason for this is that on most frames of the simulation most of the actors will not have been observed, so their positions will only have been predicted. However, under normal circumstances this should not have too great an effect on accuracy: as long as the velocity of an actor has not changed much since it was last observed its position can be calculated fairly accurately; as observations are added by the eye-gaze movement agent the mental model will be refreshed with new data, and any inaccuracies can be corrected.

There are a number of situations when more significant inaccuracies will appear:

1. The situation mentioned above of the subject concentrating on one or two other actors, perhaps while it is avoiding them, so that it does not notice other nearby actors. Or, if they have been seen they will only be monitored infrequently as a lower priority target, and their velocity may have changed significantly since the last observation.

2. Similarly, if the subject is distracted by an event occurring somewhere else in the scene then it may not pay attention to what is happening in front of it; it may therefore not notice potential collidees.

3. If an actor is occluded by obstacles or by other actors then it will not be added to the mental model in the first place, and the situation of example 1, above, of two actors colliding at the corner of a building, will arise.

4. While an actor is out of sight it cannot be monitored, so changes in velocity cannot be observed, and its predicted position will be based on the last known velocity.

5. If a large number of actors are being monitored at the same time then each actor will only receive infrequent looks, allowing more time for inaccuracies to accumulate before they are corrected by fresh data.

However, these various inaccuracies should not be thought of as a *problem* – rather they are a deliberate part of the design of the attention system. To take an obvious example, in the fourth listed case the subject should not be able to know accurate positions of the actor while it is out of sight, but it will try to predict where it might be, so that when it reappears from behind the obstacle it will be able to look for it again in roughly the right place. The fifth example demonstrates one of the fundamental aspects of the model – that not all actors can be observed at the same time.

Various features exist to counteract the inaccuracies that are inherent (by design) in the system:

- Actors for which imminent collisions are predicted are looked at frequently so that any inaccuracy is kept to a minimum (section 4.3.2.2 [Monitoring]).

- Actors have the ability to make explicit gaze requests to ensure the accuracy of information about a particular actor (section 4.3.1.4 [Actor requests]).

- Actors can look in a particular direction to make sure they know about all actors in that direction (section 4.3.1.4 [Direction Requests]).

Further discussion of the effects of using the mental model will take place in section 6.3.3.

4.3.5.3 Results

A full evaluation of the attention and mental model system is given in sections 6.2 and 6.3, after the model of behaviour has been explained; this allows quantitative data and examples to be presented.

4.4 Summary

This chapter has described the set of communicating agents that make up the attention system of each virtual actor, covering the various types of attention request agent, the eye-gaze movement agent, and how all of the agents are controlled by the attention manager. The way in which the information acquired from vision is used by the mental model was explained, along with a comparison to alternative methods and a discussion of the advantages and disadvantages of this approach.

Chapter 5

Behaviour Simulation

The previous chapter described a vision and attention system that can be used to re-create some aspects of human perception. This chapter discusses how the system can be combined with a set of behavioural rules to produce a realistic simulation of human behaviour.

An overview of the system is shown in figure 5.1. It is the job of the behaviour controller to decide what the actor should do on each frame of the simulation. Each actor is given a route to follow towards a *goal* (section 5.4.2.3); the route is made up of closely-spaced *sub-goals* (section 5.4.2.4), which are sent to the behaviour controller upon request. While following this route the behaviour controller performs *collision prediction* (section 5.1) and, if necessary, *collision avoidance* (section 5.2), based on the information obtained about the collision. Once a collision has been avoided, information about the actor's original path is used to *return* to the route towards the goal (section 5.3). The only information about the positions and velocities of other actors comes from the *mental model* (section 4.3.4).

Behavioural rules relating to collisions will be discussed first, followed by a brief explanation of goal selection and route-planning techniques.

5.1 Collision prediction

Collision prediction is an essential activity that must be performed on each frame of the simulation by every actor: if an actor does not know that a collision may happen, it cannot possibly avoid it.

Real people do not consciously have to perform collision prediction; the skill of determining, based on a person's direction of travel, whether they are on a collision course is developed through years of experience, until it becomes a

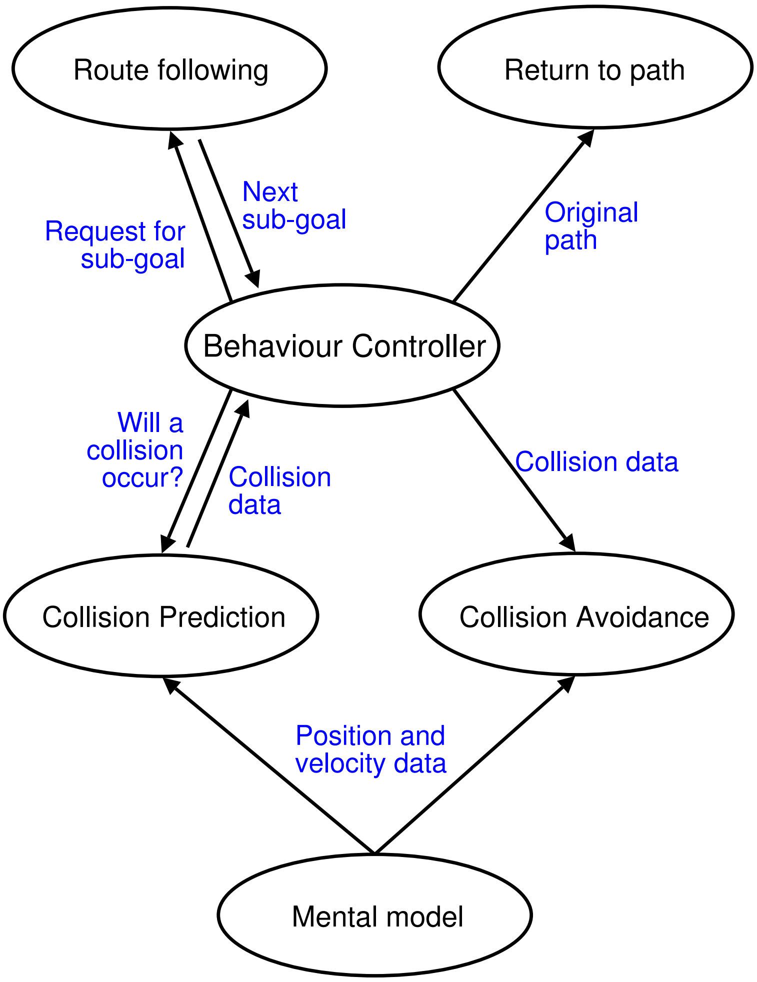

Figure 5.1: Behaviour overview

natural part of walking behaviour. However, it is clear that the process takes place, and makes use of known positions and velocities of others.

In this work, the input for the collision prediction system is the mental model of the current subject, not the actual positions and velocities of the actors (see section 4.3.4). Therefore, only actors which have actually been observed by the subject are checked for collisions, as these are the only actors for which position and velocity information is known.

For the purposes of this simulation, actors are represented by a centre, a radius and a velocity. This is an adequate representation because the circle defined by these re-creates the idea of personal space around a person. The concept of personal space is that there are rules of etiquette which dictate that people do not, under normal circumstances, walk within a certain distance of others. Goffman [Gof72] describes personal space as “the space surrounding an individual where within which an entering other causes the individual to feel encroached upon, leading him to show displeasure and sometimes to withdraw."

The exact shape and size of the personal space in which other people should not walk depends on many different factors:

- Culture – the distance at which people stand when talking to each other varies between cultures (leading to some problems with international relations).
- One's relationship with the other person – people who are friends will walk closer together than strangers would.
- The situation – people at a concert, on a train, or trying to get into a football stadium will often be closely packed together, and the usual rules of personal space will be ignored, even to the extent of body contact being acceptable.
- Personal preference – some people will try to maintain a larger area of personal space around themselves than others, perhaps if they are shy or withdrawn.

In this simulation the size of the personal space will be fixed for each actor, although it can vary between actors to simulate different cultures or personal preferences. The centre of each actor's personal space is its face: this means that the region extends further in front of the actor than behind. As Goffman puts it: "the spatial demands directly in front of the face [are] larger than at back." The reason for this is that people care more about the space in front of them than behind them – partly because this is their direction of travel and so people will obstruct their desired path, but also because people that cannot be seen can be ignored more easily. The region of personal space is shown in figure 5.2.

However, personal space can be 'invaded' if the situation demands it; in a high density crowd personal space will be ignored, as shown in figure 5.3, and the only restriction will be not allowing actors to actually pass through each other.

Predicting whether a collision will happen is not simply a case of finding out if the vectors representing the movements of two actors' centres will intersect at a shared time value: this would only work if each actor was a single point, rather than a circle of space. Rather, it is necessary to work out if a pair of actors' personal space circles will overlap at any point in the future, representing an unwanted invasion of personal space.

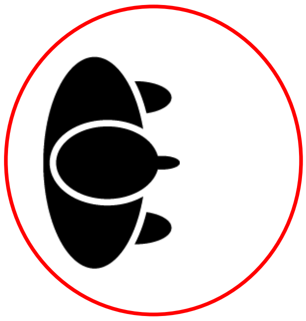

Figure 5.2: Top-down view of an actor, showing its region of personal space, centred on its face

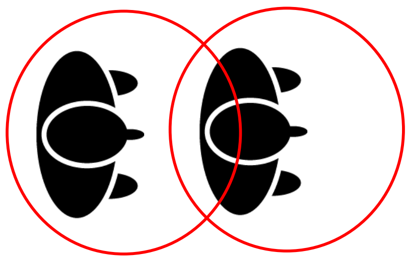

Figure 5.3: Two actors whose personal spaces overlap, but who are not actually colliding

For each of the actors in the subject's mental model the following calculation is performed to check if a collision will occur.

5.1.1 Calculation

The calculation of whether a collision will happen involves the use of the relative positions and velocities of the subject and a potential collidee. In the following equations $\mathbf{p}(t)$ is their relative position at time t, and $\mathbf{v}(t)$ their relative velocity (these are clearly found from $\mathbf{p}_1(t) - \mathbf{p}_2(t)$ and $\mathbf{v}_1(t) - \mathbf{v}_2(t)$).

The relative position varies over time:

$$\mathbf{p}(t) = \mathbf{p}(0) + \mathbf{v}(t).t \tag{5.1}$$

Therefore the distance between actors, $|\mathbf{p}(t)|$, is defined as:

$$|\mathbf{p}(t)|^2 = |\mathbf{p}(0)|^2 + 2\left(\mathbf{v}(t).t \cdot \mathbf{p}(0)\right) + |\mathbf{v}(t).t|^2 \tag{5.2}$$

which becomes:

$$|\mathbf{p}(t)|^2 = |\mathbf{p}(0)|^2 + 2t\left(\mathbf{v}(t) \cdot \mathbf{p}(0)\right) + t^2\,|\mathbf{v}(t)|^2 \tag{5.3}$$

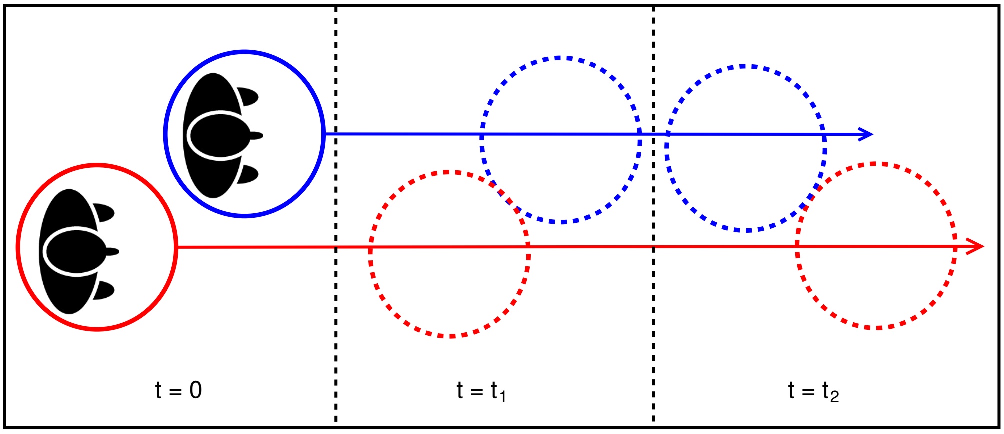

Figure 5.4: The collision between two actors starts at $t = t_1$ and ends at $t = t_2$

For a collision to occur the distance between the centres of actors must be less than the sum of their radii. The solution of the equation:

$$|\mathbf{p}(0)|^2 + 2t\,(\mathbf{v}(t) \cdot \mathbf{p}(0)) + t^2\,|\mathbf{v}(t)|^2 = (r_1 + r_2)^2 \tag{5.4}$$

returns up to two solutions, t_1 and t_2 (with $t_1 < t_2$), which are the times at which the distance between the centres of the actors is exactly the sum of their radii; these are the transition points at which a collision starts or ends. These times are shown in the example in figure 5.4.

The solutions of this equation can be divided into several cases, of which only the first two are interesting:

- $t_1 \geq 0$: a collision that will happen after t_1 frames have passed.
- $t_1 < 0$ and $t_2 > 0$: a collision which has been happening for $-t_1$ frames, and will end after t_2 more frames.
- $t_1 < 0$ and $t_2 < 0$: a collision that started and ended in the past, about which nothing can now be done.

If either of the first two cases occurs the subject stores the value of t_1 as the Time To Collision in its mental model data about the actor; it also stores the predicted position of the collision. The Time To Collision is stored even if it is negative, so that this information can be used by the behaviour controller. The procedure is then repeated for all the actors currently in the subject's mental model.

When all the calculations have been carried out, the next step is to work out which collision needs to be dealt with most urgently; this is *not* the actor which is nearest to the subject, but the actor which has the smallest Time To Collision value, as a nearby actor may be walking more slowly than an actor that is slightly further away – the more distant actor therefore has to be dealt with first.

Finally, the smallest (or most negative) Time To Collision value found is tested to make sure that the distance between subject and collidee is less than 15 m; if it is further away than this there is no need to avoid it yet, although the Time To Collision is still recorded. The value of 15 m is based on the distance given by Wolff for when collisions start to be avoided (section 3.1.3); it will be varied based on the mood of the actor – see section 7.3.2.3. If the distance is less than 15 m, the actor with the smallest Time To Collision is stored as being the current collidee for the subject, and collision avoidance will be started (section 5.2).

5.1.2 Single collision prediction request

There is an additional function, based on the main collision prediction routine, that is used to check for a potential collision between two specified actors. It uses exactly the same equation (5.4) to predict collisions, but only applies it once, checking if the subject will collide in the future with the specified actor.

The key difference is that in addition to being passed the unique *id*s of the two actors to be tested, the function is also passed test velocities for each actor. These are used in the calculation in place of the actual velocity of the subject and the velocity of the potential collidee held in the subject's mental model.

The purpose of this function is to facilitate the process of collision avoidance by allowing alternative strategies to be tested, to see if they will avert the collision – in particular changes of direction or speed; for details see section 5.2.3.

The return value is simply a boolean value representing whether a collision would occur with these values.

5.1.3 Multiple collision prediction request

During collision avoidance it is sometimes necessary to perform collision prediction between the subject and all other actors in its mental model, with certain parameters altered; this is performed by a variant of the main collision prediction routine.

The only real difference from normal collision prediction is that the velocity of the subject is replaced by a parameter passed to the function – the velocities of all the other actors in the mental model are left unchanged. For details of the uses of this function, see section 5.2.3.

The function returns a boolean indicating whether a collision would occur; if it would, then the *id* of the collidee with the smallest Time To Collision, and the Time To Collision value itself, are also returned: these will be helpful for collision avoidance planning.

5.2 Collision avoidance

If a collision has been predicted then the system tries to alter the behaviour of the subject so that the collision will not occur; in particular it will change the subject's velocity, either the speed, or the direction of travel, or both.

Avoidance behaviour can start as soon as the collision is predicted, as actors more than 15 m away will be ignored by the collision prediction routine, in accordance with Wolff's observations (section 3.1.3). However, it is rarely the case that collisions will be avoided so far off, as there will normally be nearer, more urgent collisions to attend to first.

An overview of the collision avoidance procedure is shown in figure 5.5.

5.2.1 Groups

The simplest, and often-used approach to collision avoidance is to avoid only one collision at a time, by finding the nearest actor on a collision course with the subject, and avoiding just that actor. However, there is a significant problem with this technique: if the nearest collidee is part of a group of actors then avoiding the collidee may put the subject onto a collision course with another member of the group; avoiding this actor may return the subject to its original course. This results in the subject oscillating between two (or more) courses and ultimately colliding with one of the members of the group. A better approach is for the subject to automatically detect if the actor it is trying to avoid is part of a group and, if it is, avoid the group as a whole.

Group detection is performed by an actor every frame before it decides what collision avoidance method to use. It examines all the actors currently in its mental model and calculates the distance between each of them and the collidee; if there would not be space for the subject to walk between the pair of actors, then they are considered to be in a group. Any such actors are added to the list of collidees stored by the subject.

When an actor is first detected as part of a group its position is uncertain, as the mental model may be inaccurate; as a result, it may incorrectly appear

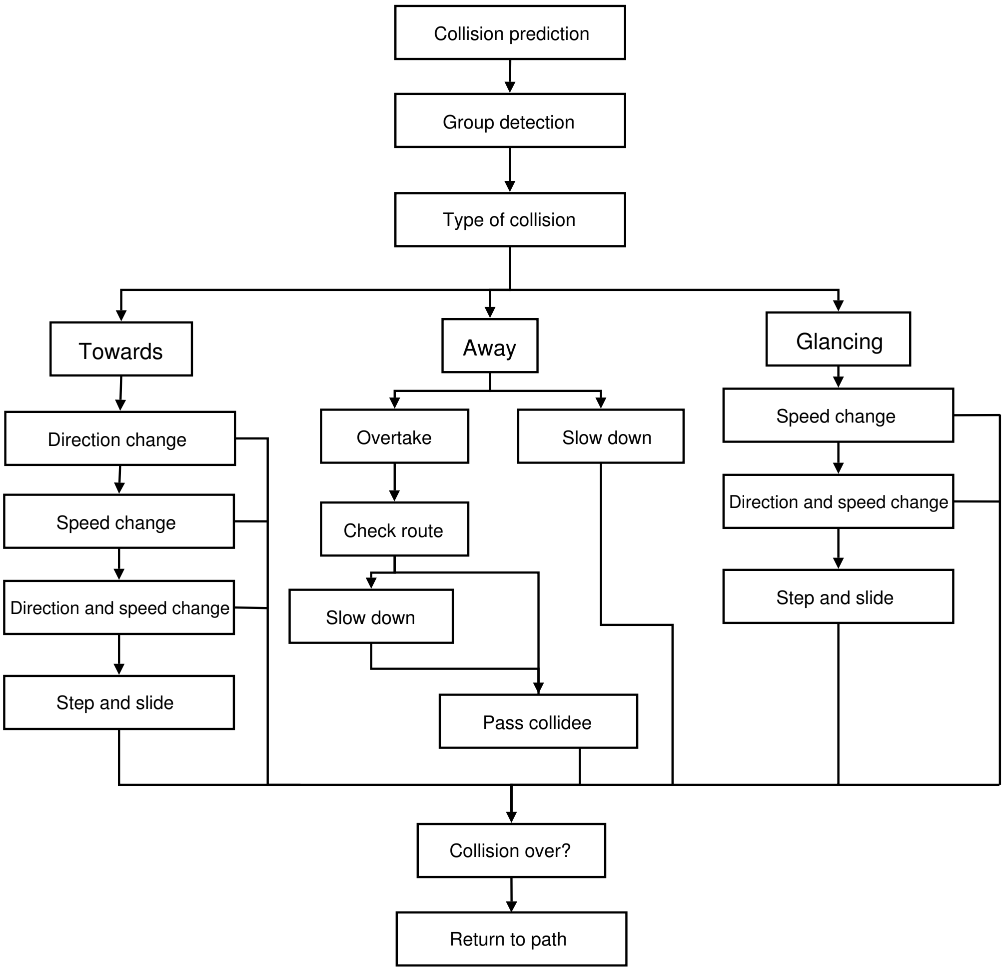

Figure 5.5: An overview of collision avoidance

to be in a group. Therefore, it is stored with a flag indicating that it has not yet been verified. At the same time an explicit request is sent to the attention manager, asking it to look at the actor and check its position. When this has been done, and the actor has been confirmed to be in the group, the flag is altered to reflect this.

The groups detected by this approach will not necessarily stay grouped together. Actors will be considered as a group even if some of them are moving in opposite directions, and are therefore only close to each other for very short periods. However, it is necessary to treat these arrangements of actors as groups: an actor trying to avoid such a collection of actors should still avoid all of them as, by definition, there are no gaps between them for it to walk through.

Figure 5.6 shows a group of actors detected by the grey actor: its collidee is coloured blue, and the other members of the group are coloured pale blue.

Because of the dynamic nature of groups, an actor must check on each frame

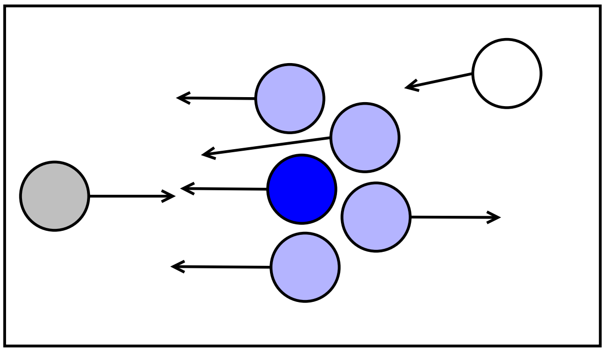

Figure 5.6: The pale blue actors are part of a group with the blue actor

that all the members of its collidee group are still in the group: if there is now space to walk between a member of the group and the collidee, that member is removed from the group.

Once group detection has finished, the centre of the group of collidees is found by averaging their positions, and the radius of the whole group is calculated. The subject will then treat the centre of the group as the location of its collidee when considering what avoidance action to pursue. As collision avoidance proceeds the specific collidee may change, but as it will still be in the same group, the location of the centre of the group will be unchanged. Therefore the subject will continue to avoid the group as a whole, rather than oscillating between different courses.

5.2.2 Collision types

If the collision avoidance routine is being called for the first time after the collision with a certain actor has been predicted, it is necessary to work out what type of collision will occur, so that the correct avoidance behaviour can be chosen. There are three types of collision, shown in figure 5.7, and given the following names in this work:

- Towards. This occurs if the two actors are walking towards each other and will have a head-on collision, either walking straight into each other, or with some smaller overlap of their bodies. Both actors should have predicted that the collision will occur, unless one of them has an inaccurate or incomplete mental model.

  This type of collision will occur if $\mathbf{v}_s \cdot \mathbf{v}_c < 0$, where $\mathbf{v}_s$ is the velocity of the subject, and $\mathbf{v}_c$ is the velocity of its collidee; in other words, the difference in bearing of the two actors must be greater than 90°.

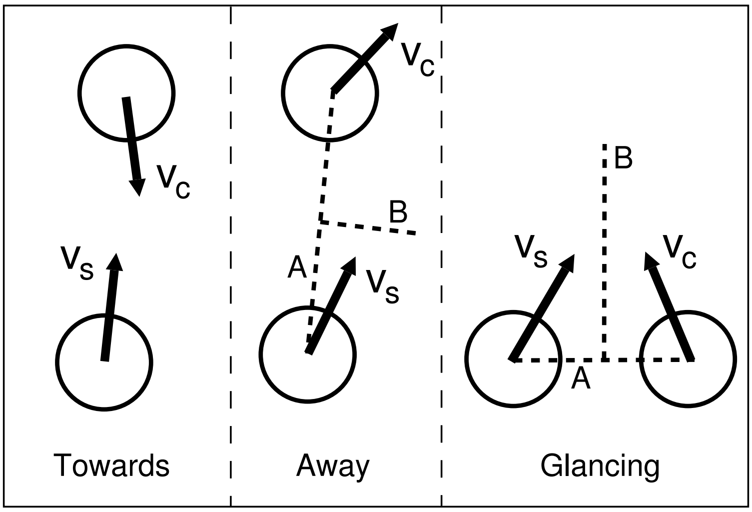

Figure 5.7: The three types of collision: Towards, Away and Glancing

- Away. Conceptually this is the situation where the subject is behind the collidee, but is gaining on it, and so would bump into its back; this is a common occurrence in the real world, as people walk at a wide variety of different speeds, and will often slow down or stop unexpectedly.

  To detect this type of collision it is first necessary to check that $\mathbf{v}_s \cdot \mathbf{v}_c \geq 0$. Next a vector between the centres of the actors is found (A), and also a vector perpendicular to this (B). If the directions of the velocity vectors of the two actors are on the 'same side' of vector B, then this is an Away collision (see appendix A.2 for a description of the routine used to perform this test).

- Glancing. This is a side-on collision between two actors that are walking in roughly the same direction, whose paths will cross in the future. The actual collision will occur sometime before the cross, particularly if there is little difference in direction of travel, as the left side of one actor will bump into the right side of the other.

  A glancing collision will occur if $\mathbf{v}_s \cdot \mathbf{v}_c \geq 0$, and if the directions of the velocity vectors of the two actors are on *opposite* sides of the vector B found above.

The type of collision is stored in the mental model so that these checks do not have to be repeated each frame.

5.2.3 Avoiding Towards collisions

To avoid an oncoming actor there are several alternative courses of action, as observed by Collett and Marsh [CM81]: change direction of travel, reduce or increase speed, or some combination of both of these. Each technique is tried until one is found that works; they are ordered in terms of increasing disruption to the subject's behaviour. Changing speed is more disruptive than changing direction as people have already chosen the speed they can comfortably walk at: walking faster will require more effort, while walking slower will hinder their progress; by contrast, a small change in direction will have little effect upon progress. Changing speed is also seen by Goffman [Gof72] as 'giving in' to the other person.

The reason for having separate checks for just changing direction and just changing speed, rather than immediately looking for a combined direction and speed change, is that this re-creates real behaviour: people prefer to make their behaviour as simple as possible, and do not want to alter their original route unnecessarily.

The four avoidance techniques are now described in detail.

5.2.3.1 Direction change

The first step in the process is working out whether the collidee is to the left or right of the subject; this is done using the whichSide routine (see appendix A.2). Section 3.1.3.1 described Collett and Marsh's observations [CM81] that if there is an overlap between the courses of two people they will always pass each other on the opposite, shorter side. If the overlap was total, they would most often pass on the right-hand side. However, it should be noted that these observations were all of situations in which there were no other, external factors that would affect the passing procedure, such as an obstruction or another person on the preferred passing side. It seems clear that if it is not possible to pass on the preferred side a person will be forced to pass on the other side, perhaps making use of the 'externalisation' described by Goffman (section 3.1.3.1) to indicate his course to the other person.

The amount of velocity direction change needed will depend on the distance between the subject and the collidee: if it is far away only a small angular direction change will be required. In any case an actor will always prefer to make only a small change in direction; this was explained by Goffman (section 3.1.3) as people finding it less demeaning to do this than having to 'give in' and make a large movement at the last moment.

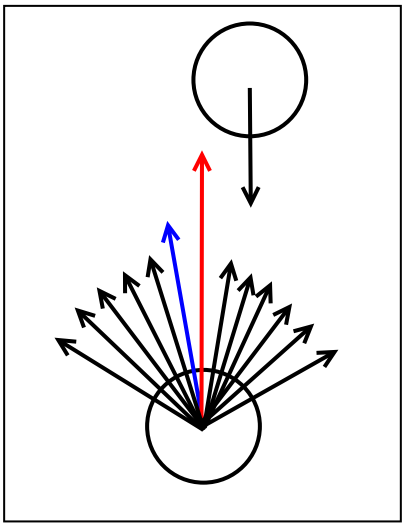

Figure 5.8: Possible changes in direction to avoid a Towards collision

Twelve different direction changes are tested, ranging from 10° to 60°, in both directions. Ideally, each actor should be able to check a continuous range of direction changes, rather than the discrete samples used by this technique. However, the only way of approximating a continuous range within this type of simulation is by testing a large number of possibilities. Twelve was chosen as a balance between getting a representative sample of directions, while keeping the process fast enough that the overall goal of a real-time system would be achievable. 10° was chosen as the smallest direction change that would allow a typical collision to be avoided.

It would be possible to calculate the smallest angle which would enable the collidee to be avoided. However, even with this approach larger angles might have to be considered due to this optimum path being blocked by obstacles or other actors. Therefore, it is more efficient for all actors simply to consider the same range of angles.

An example is shown in figure 5.8 (with the actor's current velocity shown in red): if the collidee is found to be to the right of the subject, then it should try to pass on the left; therefore the first test is a 10° shift to the left – shown by the blue arrow. If this fails, the next test is 10° to the right, then 20° to the left, and so on; all of these possible tests are shown with black arrows. Actors prefer to make a small direction change, rather than necessarily passing on the shorter side, but that will always be tried first.

To test each direction change a vector is calculated to represent the velocity the actor would have after that change. This vector is then passed to the Multiple Collision Prediction Request function (section 5.1.3), to see whether any collisions would happen if this change were applied. If no collisions are predicted then that is a suitable direction change, subject to the obstacle check described below. If collisions are predicted then the Time To Collision for the soonest one is examined: if it is more than 20 frames later than the time of the collision trying to be avoided currently, then the decision is made to take that course as it is preferable to the more imminent collision. The period of 20 frames (one second) should allow time for the subject to avoid the new collidee after it has finished avoiding its current collidee. This solves the fundamental issue that whatever direction the subject tries to move in will probably put it on a collision course with at least one actor; fortunately the majority of these collisions are many frames in the future. The chosen figure of 20 frames tries to balance the competing requirements of finding a way to avoid the current collision with not causing problems by triggering a new collision.

During the process of checking for suitable direction changes a further check must be made to ensure that the proposed direction of travel will not cause the actor to walk into an obstacle, such as a wall. To do this, the function finds the position where the actor would be 20 frames after the time of the collision, travelling in the proposed direction. The path between there and its current position is then checked, using the same code used in the vision process to determine occlusion by obstacles (section 4.3.3.1). 20 frames is once again chosen as a balance between allowing collision avoidance to happen and not wanting actors to come close to colliding with a wall.

(Collision avoidance routines are run on every frame of the simulation while a collision is still predicted; therefore as soon as an actor gets within the 20 frame limit an alternative course of action can be selected.)

If a suitable direction change has been found that will not cause a collision with another actor and does not cause the actor to walk into an obstacle, it is selected as the behaviour the actor will perform. Otherwise, the next avoidance method is tried.

5.2.3.2 Speed change

A range of different changes of speed is investigated, ranging from a 50% decrease in the current speed to a 50% increase. Small speed changes of 10% either way are tried first, followed by increments of 10%, alternating between speeding up and slowing down.

As stated above, it is clear that small speed changes are better than large changes, due to their less disruptive nature. There seems to be no evidence in the psychology literature as to whether people prefer slowing down or speeding up; personal experience suggests that people are just as likely to walk faster to get out of the way of somebody else as they are to slow down to let the other pass. Therefore, no priority is given to either method (the choice is made randomly), only to the amount of speed change.

Once again, a representative sample of discrete values is used to check the continuous range that a real person would consider, balancing comprehensiveness and realism with efficiency.

To check if a particular speed change will cause the collision to be avoided, the new calculated velocity is passed to the Single Collision Prediction Request function. This is different from the check used above for direction changes, which had to check for multiple collisions. The logic behind this is that a change of speed is very unlikely to cause a collision with another actor:

- If the subject slows down then it could possibly be bumped into by an actor walking behind it; however, this actor would not be visible, so it would be impossible to check for this type of collision.

- If the subject speeds up then it could collide with an actor in front of it; however, this is almost certain to be the actor which it is already trying to avoid, as that is the actor for which the soonest collision has been predicted.

Therefore it makes sense only to check for collisions with the single actor that is currently being avoided, as this is considerably faster than checking all possible actors. In the rare cases where this avoiding action does trigger a potential collision with a different actor, this will be detected within a few frames, and can then be avoided as normal.

If a suitable speed change is found, it is selected as the behaviour to perform. Otherwise, the next avoidance method is tried.

5.2.3.3 Direction and speed change

This is the least desirable method, as it will involve significant disruption to the actor's path, and would require more active thought by a real person. However, it is sometimes the only way that a collision can be avoided. For example, for an actor that was strolling along to avoid a fast-moving oncoming actor, it may be necessary for it to walk quickly to the side to get out of the way.

Actors will try to make small speed changes at the expense of making larger direction changes, for the same reason that changing just direction is better than changing just speed: changing speed is more disruptive than changing direction. Priority is given to passing on the shorter side of the collidee (i.e. on the left if the collidee is to the right of the subject). Therefore the order of tests is:

1. Direction change of 10° to pass on shorter side, slowing down by 10%.
2. Direction change of 10° to pass on shorter side, speeding up by 10%.
3. Direction change of 10° to pass on longer side, slowing down by 10%.
4. Direction change of 10° to pass on longer side, speeding up by 10%.
5. Increase the direction variation by 10° and go back to step 1, up to a maximum of 60°.
6. Increase the variation in speed by 10%, reset the direction variation to 10°, and go back to step 1, up to a maximum of 50%.

As each of these configurations includes a change in direction, it is necessary to use the Multiple Collision Prediction Request function, to make sure that collisions are not caused with close-by actors.

As soon as a change is found that successfully avoids the collision, and that doesn't cause the actor to walk into an obstacle, the tests are stopped, and that behaviour is selected.

If no suitable behaviour has been found, then the only option left (other than stopping) is to perform a *step-and-slide*.

5.2.3.4 Step-and-slide

The step-and-slide manoeuvre was described in section 3.1.3.2, based on work by Wolff [Wol73]. It is a way of avoiding an actual physical collision in certain situations where a detour is not sufficient:

- A detour was started, but proves insufficient for the purposes of avoiding the collision.
- The collision was predicted late, due to an incomplete mental model, and there is no time to detour.
- No suitable detour can be found due to other actors surrounding the collidee and blocking any ways around it.
- The subject chooses not to perform the detour, preferring to step-and-slide; this is based on his mood, see section 7.3.2.5.

Under normal conditions, when both actors have predicted that the collision will happen, both of them will perform the manoeuvre; this is in agreement with results from Wolff [Wol73] and Collett and Marsh [CM81], who observed that both people involved will usually cooperate to avoid any contact from taking place. (However, Collett and Marsh observed a few interactions where only one person would perform the step-and-slide; in these cases it was less successful, and some amount of collision still took place.)

There are two forms of the step-and-slide: one in which the left shoulder is brought forwards, making the body turn to the right, and one with the right shoulder brought forwards, with the body turning left. The decision about which to use is based on which side of the subject the collidee is on: if the collidee is to the right, then the left shoulder is swung forwards, as this creates more room on the right hand side of the subject. This means that actors will always perform an open pass, turning to face the actor they are avoiding; this is much easier to simulate graphically than the closed pass, which would require a severe contortion of the actor's body. The type of step-and-slide has no effect on the ability of an actor to avoid a collision, so the use of just the open pass only affects the appearance of the simulation, not the quantitative results.

The step-and-slide is triggered in the simulation when the actors are on a collision course, the time to collision is ten frames or fewer, and no suitable detour can be found that would avoid the collision in time. The manoeuvre has to be started at this point as it takes a few frames for the shoulder to swing round, and the actors should not reach each other before this has happened, otherwise they will still collide. The time to start is based on time to collision rather than distance, as this is a much better measure of the time available to prepare for the collision. Wolff states that the manoeuvre starts when the distance between people is "about 1.5 m"; assuming a frame-rate of 20 frames per second and a walking speed of 1.5 m/s for each actor, giving a relative speed of 3 m/s, this translates to 10 frames.

During the step-and-slide procedure actors are forced to pass through each other's personal space – the circles around each actor. This is acceptable behaviour that also happens in the real world. However, the condition remains that there should be no, or very little, actual physical contact, the only exception being some brushing of shoulders or arms. Therefore it is necessary to still perform the check for actual collisions, described in section 5.2.6.2, and to take corrective action if, for some reason, the step-and-slide proves insufficient.

The step-and-slide is generally of a fixed length: it takes a few frames to turn the shoulder, the actor steps forward with its body angled, then the shoulder is brought back, and normal walking resumes. Under most circumstances the step forward should be enough to take the actor past its collidee; however, if either or both actors are travelling particularly slowly, the interaction may still be ongoing: in these cases the actor may make another step while keeping its body angled. Another special case occurs if, soon after the actor has started the step-and-slide, the collision stops being predicted, usually due to some unexpected action on the part of the collidee; in this situation the actor does not go through with the manoeuvre, rather it brings the shoulder back and immediately resumes normal walking.

5.2.4 Avoiding Away collisions

To avoid an Away collision, where the subject is approaching the collidee from behind, there are two possible courses of action, described in section 3.1.4: slow down to the same speed as the collidee and walk behind it, or try to overtake. The decision about which of these to do will later be based on the mood of the actor (section 7.3.2.4); as an interim measure, each actor is given a simple variable between 0 and 100 indicating how much of a hurry it is in, and therefore how often it is willing to slow down. A random number between 0 and 100 is generated: if it is smaller than the actor's decision variable then it will choose to try to overtake; therefore, the more hurried the actor is, the more likely it is to try to overtake.

The decision about overtaking is made when avoidance behaviour for a particular collidee starts. If the collidee changes, a new decision is made; otherwise the same course of action is kept until the collision threat is over.

The two forms of avoidance behaviour are now discussed.

5.2.4.1 Slowing down

If the decision is made to slow down then the actor first checks that the collidee is actually moving: if it is not, then it makes no sense to slow down to its speed – i.e. stop – so it should overtake instead.

The next action taken is to set the velocity magnitude to be the same as that of the collidee. The magnitude is stored as a separate scalar value from the velocity, to simplify returning to the original velocity when the collision has been avoided, and is applied to the velocity before it is added to the actor's position each frame.

Slowing down is not always sufficient: if the collision was not predicted until late on then the subject may be almost bumping into the back of the collidee. Personal observations show that people do not usually walk this closely behind other people (unless, perhaps, they are in a group and know each other); doing so is likely to cause discomfort, anxiety or anger, as well as making a collision very likely if the collidee slowed down further. An exception to this rule occurs if the crowd is very densely packed, such as at a stadium, in which case people may be packed closer together.

The limit used in this simulation is a 1 m gap between collidee and subject: people were rarely seen walking closer than this in personal observations. If the subject is nearer than that then its velocity magnitude is set to be $\frac{2}{3}$ of the speed of the collidee: this means that over the next few frames the distance will steadily increase, until it reaches 1 m, at which point the velocity magnitude can be reset to the speed of the collidee.

The other restriction on walking behind another person is that one should not walk *directly* behind them; as Wolff [Wol73] (section 3.1.4.1) described it, the subject will "strive to maintain a head-over-the-shoulder relationship with the person in front" if the distance between them is less than 5 m. This serves two purposes:

- It allows the subject to see over the shoulder of the collidee, to check for any oncoming people who might need to be avoided.
- It follows social conventions: walking directly behind somebody is more likely to cause them discomfort than walking slightly to one side.

An actor is directly behind its collidee if the angle between the collidee's velocity (shown with a black arrow in figure 5.9) and the vector between the two actors is less than 10° – shown by the angle α.

The actor should move the shortest distance possible to correct this situation, for example if it is already slightly to the left of the collidee, it should move further left; the whichSide routine (appendix A.2) is used to determine whether the subject is to the left or the right of the collidee. Once the direction to move has been determined, a vector perpendicular to the subject's velocity, with a magnitude only 20% as large, is added to the velocity in the appropriate direction; the combined velocity is shown by the blue arrow in figure 5.9. This will cause the actor to slowly move sideways, in addition to its forward movement, until it is slightly more than 10° to one side; it then resumes its original velocity.

The subject will continue to walk at the same speed as the collidee until conditions change:

- If the collidee slows down, the subject will slow down too, to keep a constant distance between them.

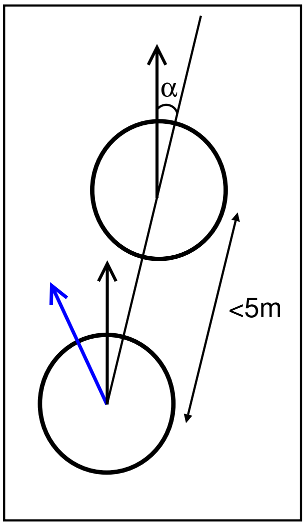

Figure 5.9: Avoiding walking directly behind the actor in front

- If the collidee speeds up, the subject will match that speed, until it gets to the speed the subject was originally walking at.
- If the collidee starts to walk faster than the subject's original speed, there will no longer be a collision predicted, as the collidee will be moving away from the subject; therefore the subject returns to its original speed.
- If the collidee changes direction such that the subject would no longer collide with it, the subject returns to its original speed.
- If the collidee stops, the subject will be forced to overtake it.

These rules combine to produce the effect that an actor has a certain speed that it prefers to walk at; it will slow down as needed to avoid bumping into people, but it will return to that preferred speed as soon as possible. This ties in with Goffman's ideas [Gof72] that people have set speeds that they will try to maintain, not walking too quickly for their liking, and not walking so slowly that they will not arrive at their destination in time, but also sometimes willing to slow down rather than have to make the effort to overtake. (See section 3.1.4.1 for a discussion of the advantages of slowing down and lane formation.)

5.2.4.2 Overtaking

If the decision is made to try to overtake the first step is checking whether in fact it is currently possible to do so, by looking at the positions of other actors in the vicinity of the collidee to see if there is any space to get past. However, it is possible that the subject does not have information about all the relevant actors in its mental model, as they may not yet have been observed. Therefore, it is necessary to check for any unobserved actors that might be of interest.

Observations To perform these observations the actor makes use of the explicit direction request agent (section 4.3.1.4). It sends a request to look in the general direction of the collidee, with a priority of zero – the lowest value. As the attention manager will look for actors within 15° of the stated direction, this will look at any actor close enough to the collidee to cause any problems with overtaking. A priority of zero is used so that any other explicit requests will take priority over this (but this will still have higher priority than other types of attention shift); this is because the decision about whether to overtake or not is not urgent – the subject can always slow down and wait if necessary – whereas other requests, such as those related to imminent collisions, may be much more urgent. The actor sends a fixed number of explicit requests, all the same; this ensures that all the unobserved actors near the collidee will be looked at. It does not matter if more requests are sent than there are actors to look at, as if the attention manager cannot find an actor to observe it will simply dismiss the request. This technique matches the human behaviour of checking until it is certain that there is no-one else to check.

After the actor has made the explicit requests it goes into a waiting state until those requests have completed; it is notified by the attention manager when the requests have been completed. While in this waiting state the actor will continue walking at its normal speed, unless it starts to get too close to the collidee. If it gets within 2 m of the collidee, then it will slow down to the same speed; if it is less than 1 m from the collidee, due to the behaviour just starting, it will slow down to a speed $\frac{2}{3}$ that of the collidee, to steadily draw back from it.

The psychological justification for this behaviour is that when a person is planning on overtaking somebody else, he will inspect the area around that person and, normally subconsciously, use this information to work out if there is space for him to get past. This brief inspection is re-created by the explicit direction requests. The waiting state represents the thinking time while a person would consider whether he can get by; in real life, as here, this time should normally be very short, and will only last for a small fraction of a second.

Preferred side Once the necessary observations have been completed the actor can decide, based on its mental model, whether it can overtake. Firstly, it looks to see which side of the collidee it is on, and therefore which is the

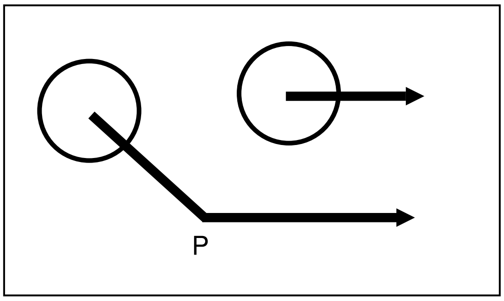

Figure 5.10: The route an actor takes to avoid its collidee

shorter side to pass on. This makes use of the same idea as when avoiding Towards collisions (section 5.2.3.1): the subject should try to minimise the distance it has to walk to avoid the collision, an effect observed by Collett and Marsh [CM81]. In other words, if the collidee is on the left side of the subject, then it should prefer to overtake on the left side.

Route check The overtaking procedure consists of the actor moving out diagonally until it is clear of the subject, then walking straight ahead to get past it, as shown in figure 5.10. The angle of the diagonal component is variable, based on local conditions – different angles are tried to find one that allows an unobstructed path, varying from 20° to 60° in steps of 10°. Two parts of the path must be checked: the diagonal part, and the straight ahead part.

The diagonal part is checked using a call to the Multiple Collision Prediction Request function, passing it the proposed diagonal velocity. If no collisions are predicted, this part of the course is fine. If collisions are predicted, then the path may still be usable if:

- The time to collision is greater than that for the collidee: this means that the subject will be off the diagonal part of the path before the new collision would occur.
- The time to collision is more than 20 frames: in these cases the actor should be able to complete the diagonal movement before this secondary collision occurs – it can be abandoned if the time drops below 20 frames. The figure of 20 frames was chosen for the same reasons as in section 5.2.3.1, as a balance between being able to avoid the current collision and not causing another.

Next, the straight ahead part of the path needs to be checked. The point at which this starts (point P in figure 5.10) is found as follows:

1. Store the subject's position, as a backup.
2. Update the subject's position by the amount it would move down the diagonal path in one frame.
3. Use the Single Collision Prediction Request function to see if the subject would collide with the collidee if it walked straight forward from there. If it would, loop back to step 2 – it is not yet far enough down the diagonal path to be able to get past the collidee. If it would not collide, its current position is point P.

Using the discovered position, a call is made to Multiple Collision Prediction Request to see if there are any potential collisions, with actors or obstacles, along the path ahead. If there are no collisions, or no collisions with a time to collision of less than 20 frames (chosen for the same reasons as above), the path is considered safe.

If either the diagonal or straight ahead part of the course is blocked then other paths can be tried. First, overtaking on the opposite side is tried; then the angle of the diagonal path is increased by 10° and the whole procedure is repeated. In other words, a small change of direction is given priority over the preferred side for overtaking; this is justified by Goffman's theory [Gof72] of people always diverting from their course by the minimum amount. However, the subject will usually overtake on the preferred side, at an angle of 20°, as under normal circumstances there will often not be any other actors blocking its path.

Speed boost If an overtaking path has been found then the chosen diagonal component is added to the subject's velocity. The subject is also given a speed boost, by increasing its speed by 25%. This is based on the observation that people will walk faster than normal while overtaking, so that the procedure can be finished quicker: while overtaking people are often out of the normal flow of traffic, or may even have moved onto the road, so they will feel 'out of place'; therefore they will want to return to their normal path as soon as possible by quickly completing the overtaking manoeuvre.

Alternative If no overtaking path has been found then the subject will wait and try again next frame. In the meantime, it may need to take some intermediary action. If the distance between the two actors is under 2 m then the subject will slow down to match the speed of the collidee. If it is under 1 m, its speed will be set to $\frac{2}{3}$ that of the collidee, so that it will steadily drop back.

This means that the subject will try to keep walking at its current speed for as long as possible: as it has chosen to overtake rather than slow down, it clearly does not want to slow down unless necessary. However, if it gets too close it will prove harder to overtake without bumping into the collidee, therefore it will slow down when required.

The actor will continue the overtaking procedure, walking with a boosted speed, until it is safely past the collidee; the conditions for this will be discussed in section 5.3.1. However, there are a number of ways in which overtaking may be curtailed or altered:

- Once the subject has moved out to the side of the collidee it may detect a collision with an oncoming actor that had not been predicted before, perhaps due to a mental model that was incomplete despite the Direction Request observations, possibly due to occlusion. It will then have to perform Towards collision avoidance on this actor (section 5.2.3).
- Alternatively, it may discover that its path around the collidee is blocked by another actor walking in the same direction as the collidee. In this case it can choose either to overtake this actor as well, by continuing on its diagonal path, or to give up now and slow down.
- If the subject is still behind the collidee (where it can continue to observe it), and the collidee speeds up or changes direction, the subject may no longer need to overtake it, as it will no longer be an obstruction to its path; it will therefore return to its original velocity.

5.2.5 Avoiding Glancing collisions

Of the three types of collisions, glancing collisions are the least common – in real life as well as in this simulation. This is due to the fact that most people walk roughly in paths that are parallel to each other, either in the same or opposite directions. For example, on pavements, most people will walk parallel to the edge of the pavement. This causes a large number of potential Towards collisions, as people approach each other head on, and Away collisions, as people choose to walk faster than others. Glancing collisions require people to be walking in roughly the same direction, but with their paths converging; this occurs only if people are moving slightly sideways across the pavement.

As a result of their infrequency, less time was spent attempting to simulate Glancing collisions. Fortunately, they can be handled in roughly the same way as Towards collisions.

The best way of avoiding a Glancing collision is to change speed: either speeding up to get to the collision point before the collidee gets there, or slowing down so that the collidee will have passed by the time the point is reached. This is re-created using the 'Speed change' routine from the Towards collision avoidance method (section 5.2.3.2).

The actor that detects the collision first will do its best to avoid the collision on its own, either by speeding up or slowing down. The other actor may then not have to alter its course at all, or may have to make the opposite choice from the first actor, i.e. slowing down if the other had speeded up.

If an actor cannot avoid the collision using speed changes alone then the combination of direction and speed changes used to avoid Towards collisions (section 5.2.3.3) is tried.

If none of these actions is sufficient to avoid the collision taking place, the actors will check if performing the step-and-slide (section 5.2.3.4) would help, although this will not be any use in most situations. In circumstances where this does not help, an actual collision will take place; such collisions are discussed in the following section.

5.2.6 Collisions

Despite the best efforts of the collision avoidance routine, sometimes a collision may actually occur, either with an obstacle or with another actor. This is because of the use of the mental model for actors, rather than a system where actors have total awareness of their surroundings. Some of the effects of the mental model approach were discussed in section 4.3.5.2. To summarise, the issue is that the subject may not be aware that another actor exists, because it has not observed it yet; even if it has seen the actor at some point it may have out-of-date information about its position and velocity, and therefore not realise it is a threat.

Collisions are not an error in the simulation; rather they are an intended feature. In the real world collisions between people happen frequently, due to people being distracted by looking elsewhere. For example, if a person is looking at a companion, reading a map, looking over his shoulder, or just looking round at the scenery, he may not notice another person walking towards him, or an obstacle such as a lamp-post.

In addition, both Collett and Marsh [CM81] and Hill [Hil84], as quoted in section 3.1.3.3, observed that small collisions occur frequently even if both people have seen each other, due to one of them not properly following the social rules necessary to avoid a collision. For example, if a person expects that the other will cooperate with him, and so moves only partially out of the way himself, a collision will occur if the other person does not do so. Likewise, as mentioned previously (section 5.2.3.4), if only one person performs the step-and-slide manoeuvre a collision may result.

While it is essential that the behaviour controller allows collisions to happen, it should not be possible for actors to actually walk through each other. Overlapping actors would not be considered a problem by the 3D engine (section 6.1.1.1), which just displays actors as collections of triangles; therefore, there needs to be a function within the simulation that acts 'out-of-model' to prevent this from happening. In other words, this function will have access to complete information about all actors, and will stop any impossible behaviour from occurring. If it has to stop any such behaviour then it will pass appropriate information to the agents for the actors in question, so that they can then deal with the situation.

Two types of collision need to be dealt with: collisions with obstacles and collisions with other actors. These functions run after the behaviour controller for each actor has set what it would like its actor's new velocity to be for that frame, and after the new desired positions of all the actors have been calculated. The purpose of the tests described below is to see if these new positions are in fact possible. Running after all the positions have been calculated means that all actors have access to the same set of data; until all the new positions have been found it is impossible to determine whether or not a collision will take place: actors may move out of a colliding position just as easily as moving in to one.

5.2.6.1 Collisions with obstacles

Detecting whether an actor would collide with an obstacle if it moved to its new position is simply a matter of checking whether that position represents a square on the map of the scene (see section 5.4.2.2) that is marked as containing an obstacle – such as a road or a wall.

If an obstacle is detected the first action is to prevent the actor from moving forward; instead its position is left the same as it was last frame. This represents the way that a real person would not be able to walk through an obstacle: he would either rebound off it if he did in fact walk into it, or he would stop at the very last moment. In the case of a small obstacle like a lamp-post the former can happen – people do really walk into them. If the obstacle is larger, such as a wall, this is less common, as it is very unlikely that they will not have seen it before they actually hit it; however, they may only see it at the last moment, and suddenly need to change direction.

The actor now has to be given a new route that does not consist of walking through the obstacle. As will be discussed in section 5.4.2.3, each actor has a goal that it is walking towards, following a route made up of a list of nodes. The function used to create the route is guaranteed not to allow it to pass through any obstacles. Therefore the route planning algorithm (section 5.4.2.4) is called to produce a new route from the actor's current position to its goal. Normally, this will result in only a small change of direction, as the actor should have been following nearly the same route before the collision.

The reasoning behind this is that if a person is so absorbed in his thoughts, looking around, or looking down, that he does not notice an obstacle until he collides with it, he will go through a period of reorientation, during which he will work out where he is in relation to where he is trying to go, and then set off in an appropriate direction. This is the effect that is re-created here.

Obstacle collision should not occur very often:

- If an actor sticks relatively closely to its planned route it will never collide with an obstacle.

- When performing collision avoidance with another actor the potential behaviours are always checked to make sure that they do not involve walking through an obstacle.

5.2.6.2 Collisions with other actors

Detecting whether an actor would collide with another actor if it moved to its new position is a more complicated procedure; it is first necessary to define what is meant by two actors colliding.

For the purposes of the collision prediction and avoidance that is performed by the actors' agents, each actor is defined by a circle of personal space, through which they should not walk. However, merely checking if the personal space circles for a pair of actors overlap is not sufficient for collision detection: although this represents an invasion of personal space, their bodies may not actually be touching; so although this situation is not desired, it is permitted. Therefore, collision detection is performed by representing actors as ellipses, which are a close fit to an actor's outline. This is a compromise between using circles, which are not accurate enough, and doing full collision detection on all parts of the body, which would be prohibitively expensive.

The first stage in the process is detecting collisions for all actors; each actor is selected in turn, and collision checks are performed with all other actors.

The shape and rotation of the ellipse representing each actor are changed if an actor is performing a step-and-slide. The mathematical detail for this is given in appendix A.1. A list of collidees is stored for each actor in a stack.

Another stack is used to store actors whose collisions have not yet been dealt with. At first, this consists of all actors, but actors may get added back into the stack if they need to be processed again. Actors are continually popped out of the stack and processed, until it is empty. Processing for each proceeds as follows:

1. Pop the first item off the stack of actors that are colliding with the subject.

2. Send an Explicit Actor Request to the Attention Manager to look at that collidee, with a priority of 100. This is the highest priority, meaning that the collidee will be looked at urgently. This re-creates the effect that if a real person bumped into somebody, he would definitely turn to see what had happened.

3. Check if the collidee is behind the subject: if it is, then the subject should walk forwards as normal, as this will improve the situation, by taking it further away from the collidee (who will stop, when it is its turn to be processed). The subject is put into a state where it will walk forwards, but this may be cancelled by the effect of other collidees on the stack in step 4.

4. If the collidee is not behind the subject, then the subject must stop: its position and velocity are set to stored values from its last position. This stops an actual collision from happening, and re-creates the real-world effect of impenetrability of people. However, this causes a problem: the actor is now in a different place from the position it was in when the collision checks were done, meaning that it could now be colliding with additional actors; it has no other movement options, so these actors will have to be moved. Therefore it must:

    (a) Iterate through all actors and check for collisions with them, using its own, new, moved-back position, storing any collidees in its stack.

    (b) For all the collidees, check if they are yet to be processed. If a collidee is still to be processed, then there is no problem, as it will make use of the new position of the subject when its turn comes: the only step needed is to add the subject to the collidee's own stack of collidees – this saves it from having to perform the calculation itself. If however the collidee has already been processed, then it must be

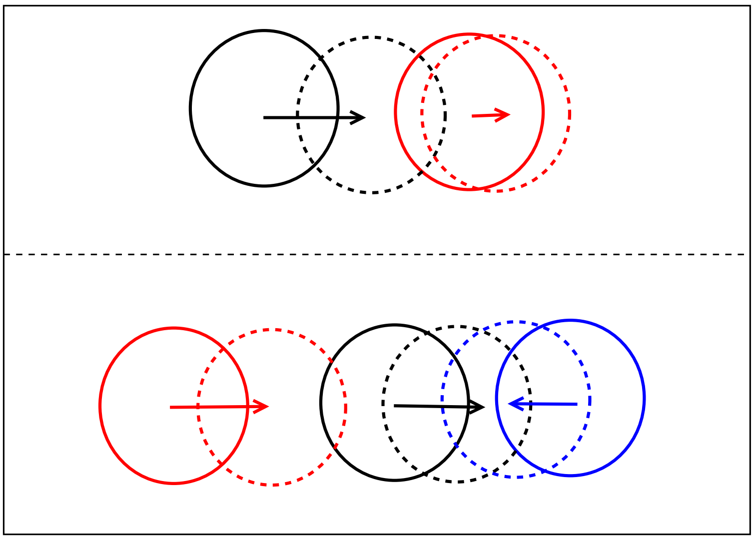

Figure 5.11: Two examples of the procedure involved in avoiding actual collisions

processed again, to take into account the new position of the subject – it may no longer be able to perform the planned action; therefore the collidee is added back to the processing stack, with the subject added to its stack of collidees. (This will never cause an infinite loop, as the routine will always terminate in the extreme case of all actors having been moved back.)

After all the actors have been processed – some of them more than once – their positions and velocities are recorded: these are values which are known not to cause collisions, and which can be used, if necessary, on the next frame, as part of step 4.

Two examples of the procedure in action are shown in figure 5.11. In the top half of the figure the black actor is catching up to the red actor, and would collide with it if both actors used their new positions. Their current positions are shown with solid circles, their new positions with dotted circles. Because the red actor is in front, it can safely walk forwards, and it does so; the black actor is forced to stop. Because of the attention request it will now be aware of the red actor, and so can take suitable action for avoiding Away collisions (section 5.2.4).

In the lower half of the figure, the black and blue actors are having a Towards collision; neither of them can safely walk forwards, so both will have to stop. However, if the black actor stops then there will be a collision between it and the red actor that would not have happened if the black actor had moved forwards as expected. If the red actor had been processed first (and found that it could walk forwards) it will need to be processed again; this time it will be forced to stop. If a cluster of actors are all walking behind each other then the effect will ripple down the line.

It is clear that in many cases this procedure will involve several actors in a collision being forced to stop, with no chance of being able to move on subsequent frames – for example, the black and blue actors in the lower half of figure 5.11. Therefore, there must be a way to break these deadlocks.

5.2.6.3 Breaking deadlocks

If an actor was unable to move on the previous frame then its velocity direction is set to point at its current sub-goal. This is a way of making sure that the actor is not needlessly stuck. This can occur if an actor has diverted from its planned route to avoid another actor; however, if it is now stuck then that avoidance has clearly failed, and so there is no point in continuing in that direction. Therefore it is worth checking if it can get out of the situation by returning to its planned path. Its velocity is also now correctly set for when it is next able to walk forwards.

If the actor is still unable to move, then it is in deadlock with another actor. These situations happen in the real world too:

- If two people nearly collide while walking towards each other then they will attempt to walk round each other by side-stepping left or right. Problems arise when both people move to the same side; this may happen repeatedly, normally to the embarrassment of both participants.
- If two people are attempting to walk through a doorway from opposite sides then one of them may have to retreat to let the other person through first.
- If somebody in front of a person suddenly stops walking then that person may have to suddenly side-step to get around him.

Collett and Marsh [CM81] (section 3.1.3.3) observed a 'pedestrian dance' or 'stutter-step' that occurs when collision avoidance breaks down or when a collision has happened. They do not define these phrases, but here they are interpreted as meaning a number of side-steps to try to get round each other.

The function proceeds as follows; if at any stage a suitable behaviour is found the function is terminated.

1. Check if the actor can now walk forwards without a collision taking place.

2. See if the actor would be able to walk forwards if it performed a step-and-slide manoeuvre, either turning to the left or to the right.

3. Check if a side-step to the left or to the right would cause a collision: if not, walk that way.

4. Try walking one step back, combined with a side-step to the left or right.

5. Finally, try simply walking back one step. This is the least preferred option as it often only postpones the problem by a frame.

The order of testing side-steps to the right or left is random, to ensure that not everybody does the same thing; however, after the first side-step, continuing in the same direction is preferred to changing to the opposite way. This also has the effect of re-creating the real-world effect of people both moving the same way and not being able to get past each other.

The effect of the above rules is that actors try the simplest techniques first – walking forwards, with a possible step-and-slide – then side-steps which have a neutral effect on their progress, before finally backing off and therefore delaying their progress; this, once again, implements the idea of people preferring actions which have the least effect on their route.

In practice, actors are normally able to break a deadlock using a small number of side-steps, followed by walking forwards and using the step-and-slide; this re-creates the 'pedestrian dance' described by Collett and Marsh. As in real life, being forced to walk backwards only occurs very rarely.

5.2.6.4 Summary

Actors in this system may collide with obstacles or with other actors, due to inaccuracies in their mental model; this is an accurate re-creation of real life, in which collisions can occur due to people being distracted. Collisions with obstacles are dealt with by giving the actor a new route. Collisions with other actors are solved by one actor stopping and the other walking forwards or, if that is not possible, by both actors performing some combination of side-steps.

5.3 Returning to path

Section 3.1.5 discussed observations by Wolff [Wol73] and Goffman [Gof72] concerning people's behaviour after they have finished avoiding a collision. They discovered that people did not behave in the way that common sense would perhaps suggest: the expected behaviour was that if a person had had to move to one side to avoid an oncoming person he would simply continue to walk forwards from his shifted position, planning a slightly different route to his goal. In the particular example of pedestrians on a pavement it would seem unimportant which part of the pavement a person would walk on, as it all leads to the same place.

However, what both psychologists noticed was that after avoiding the collision, people would in fact return to the path that they had originally been following, as if they had never had to move off it in the first place.

Most previous simulations have taken the common sense approach; for example, in Thompson and Marchant's system [TM95] once an actor has successfully made its way around another, it will conduct route-planning behaviour to work out a new optimal route to its goal from its current position.

In this work the real-world technique is simulated. The problem can be divided into several areas:

- Determining when the collision has been successfully avoided.
- Checking if it is possible to return to the previous path without a sudden change in direction.
- Planning a new route to the goal if the previous step was unsuccessful.

These areas are now explained in detail.

5.3.1 End of collision avoidance

Determining the end of collision avoidance requires different approaches depending on the type of collision.

Towards collisions The most obvious way for a collision to be over is if the collidee is now behind the subject – the subject has walked past the collidee; it is therefore now safe for the subject to return to its path. The real-life basis for this is clear: once a person has walked past whoever he is avoiding he can no longer see them, and they are no longer a threat. In the example shown at the top of figure 5.12 when both actors are at the positions shown with dotted circles, collision avoidance is over.

A second check is also needed: if the collidee is now more than 4 m away from the subject, and the collision is no longer predicted, then the collision is considered to have been successfully avoided. This ensures that if the collidee has changed direction and will no longer collide then the subject will stop trying to avoid it; the distance check means the subject has to wait until it is certain

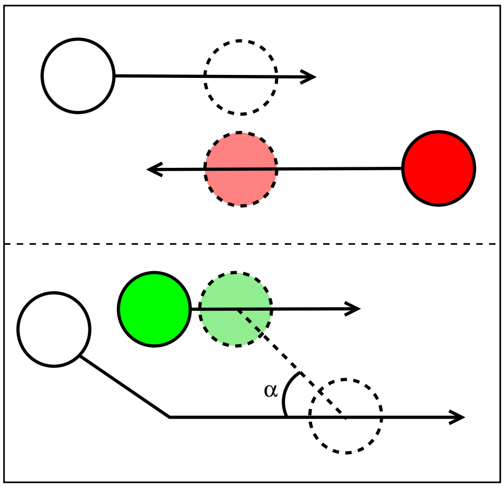

Figure 5.12: Determining when collision avoidance is over for Towards and Away collisions

that the collision is over, rather than just a minor fluctuation in the collidee's course.

Away collisions The situation for an Away collision, when the subject has overtaken the collidee, is a little more complicated: if the subject were to return to its path as soon as it was in front of the collidee, it could have a Glancing collision with it; even if this does not happen it would end up walking far too close in front of the collidee.

An example of how this problem is solved is shown in the lower half of figure 5.12: the subject uses its mental model data to calculate the relative position and velocity of itself (shown in white) and its collidee (in green), and from this finds the angle between them, shown as α; if this is less than 45°, as it is here, then it is now safe for the subject to start moving back to its path.

Alternatively, as before, if the subject is now more than 4 m away from the collidee, with no collision predicted, then avoidance behaviour can be abandoned.

Glancing collisions A Glancing collision is treated in the same way as a Towards collision: collision avoidance is over if the collidee is now behind the subject, or if the distance between them is more than 4 m. The former represents the subject and collidee having 'decided' who is going to go first – that actor will have walked faster and got in front; both actors can then go back to their previous paths.

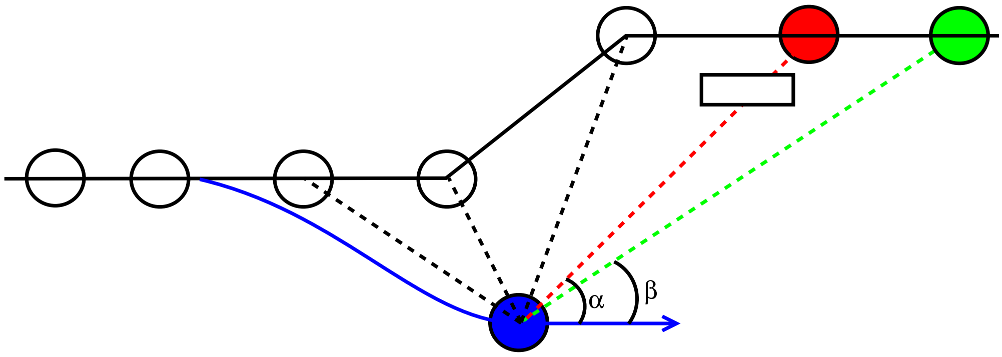

Figure 5.13: Checking the sub-goals to find a way back to an actor's original path

5.3.2 Considering path ahead

It has already been stated that actors should attempt to return to the route they were previously following before the collision avoidance behaviour was started. However, it is also clear that actors should not suddenly change direction in an attempt to get back to this route as soon as possible, as this would look artificial. Rather, they should gradually return to their route by changing their direction of travel so that it will finally intersect with the route, at which point they will carry on as before.

Each actor's route consists of a number of sub-goals, leading to its final goal. The task is therefore to find which of the sub-goals is a suitable target to walk towards; once the actor gets to the sub-goal it will continue walking down the route as normal. The vectors between each sub-goal and the actor's position are calculated, and the angles between these and the actor's current velocity are found. The earliest sub-goal for which the angle is less than 45° is chosen. The further the actor has had to stray from its route, the more likely it is that this sub-goal will be late on in the path; for example, if the actor has been overtaking several actors, it will have travelled forwards significantly, walking past several of the sub-goals while doing so.

An example is shown in figure 5.13. The actor's set of sub-goals are shown by circles; it is currently at the blue circle, having followed the blue line, away from the black planned path, due to collision avoidance. It considers all the sub-goals that were ahead of it when it left its route – dotted lines are shown between its position and each of these sub-goals. The first of these sub-goals that gives an angle (shown here as α) of less than 45° is shown with a dotted red line and a red circle.

Once a sub-goal has been chosen, it is necessary to check whether the actor can walk to it without walking through an obstacle (such as a wall or a road).

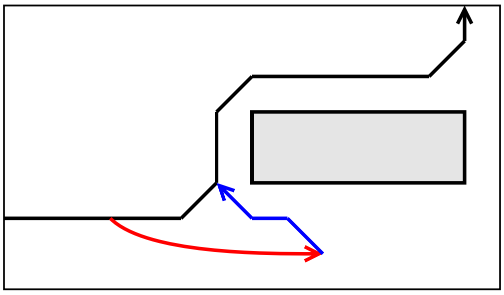

Figure 5.14: Choosing a new path, most of which is the same as the old path

The original route is guaranteed to be obstacle-free (see section 5.4.2.4), but once the actor has diverted from the route there is no guarantee that getting back to it will be safe: in figure 5.13 the dotted red line passes through an obstacle. The routine for obstacle occlusion originally written for the visual system (section 4.3.3.1) is used to perform this check, testing if the 'line of sight' between the actor and the chosen sub-goal is free of obstacles. The only change is that obstacles that would not block vision (such as roads) will block the potential route. If the sub-goal is found to be unsuitable, the process continues with later sub-goals; in this example, the next sub-goal, shown in green, is found to be suitable: the angle β is less than 45°, and there are no obstacles in the way.

5.3.3 Choosing new path

It is possible that none of the sub-goals will be found to be suitable, particularly if the actor has diverted a long way from its path. For example, if it was supposed to be turning at the corner of a building, but missed that corner due to avoiding another actor (who was blocking its planned route), the sub-goals will all either be behind it, or may be blocked by the building. If this is the case, then the actor simply plans a new route from its current location to its final goal (section 5.4.2.4); this will normally share a large number of sub-goals with the original route. In the example shown in figure 5.14 the actor has diverted from its original path (shown in black) down the red line, to avoid a collision. Considering the sub-goals ahead of it, they either do not meet the 45° requirement, or are blocked by the obstacle. Therefore it chooses a new path, shown in blue; most of the path overlaps with the original (black) path.

Finally, if the actor is found to be already in the same square of the map as its goal, it should just walk straight towards the goal. The size of each map square is 2 m, so this is a good representation of Wolff's statement that a person would walk straight to his goal if he were just a 'few metres' from it.

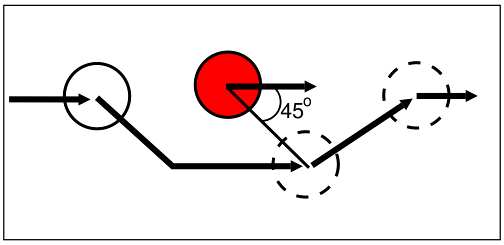

Figure 5.15: The overall Return to Path process, for an Away collision

5.3.4 Other considerations

When the collision avoidance behaviour is stopped it is necessary to:

1. Cancel any speed boost given to the subject to help it finish avoiding the collision quickly: this only applies to Away collisions. The subject continues walking at the boosted speed until it is at the 45° angle described above; after this it is safe to slow down to its normal speed, which will still be faster than the collidee's.

2. If a step-and-slide manoeuvre has just been started a few frames ago, and the shoulder is still being angled around, cancel the manoeuvre by bringing the shoulder back immediately: as the collision is over, there is no need to perform the full action.

5.3.5 Summary

In normal operation the most frequently occurring action is that the actor will choose a sub-goal one or two ahead of the sub-goal it was walking towards before. This will produce a smooth overall motion in which the actor moves out, either to avoid or to overtake, and then gradually moves back to its path, as shown in figure 5.15: this is a good re-creation of the effect described by Goffman and Wolff. It is rare that a new path needs to be created, but the option is present if necessary.

5.4 General behaviour

While Collision Avoidance is the most important aspect of behaviour in this work, it is clearly necessary to give actors activities to perform when they are not actually avoiding a collision. Not only does this make them look more realistic, but it brings actors into contact with each other, generating more possibilities for collision avoidance.

Two aspects of general behaviour are discussed here:

1. variation in speed of walking;

2. generation of routes for actors to follow.

5.4.1 Walking speed

Results of studies into the speeds people prefer to walk at were discussed in section 3.1.2. The most important result was that the speed ranges from a minimum of 0.86 m/s to a maximum of 2.18 m/s [DH03]. This range is re-created in this system, having first been converted to metres per frame, using a value of 20 frames per second.

For now, each actor is simply given a random value within this range. However, in section 7.3.2.1, techniques for varying the speed based on the actor's mood will be discussed.

The value selected acts as the actor's *preferred* walking speed, and will be used whenever the actor is not avoiding a collision; during avoidance behaviour the speed may change, as detailed throughout section 5.2.

5.4.2 Routes

The process of route planning and route following can be broken down into a number of areas:

- choosing goals;
- planning a route to get to each goal;
- following the route and behaviour having reached the goal;
- on-the-fly route planning.

Goals are chosen for each actor, and routes planned between these goals are calculated, in a processing step before the actual simulation is started. This means that the moderately slow process of route planning will not affect the speed of the simulation, unless an actor needs to plan a new route to a goal (section 5.3.3), or an actor runs out of goals. Currently the number of goals is set at ten, which enables most simulations to be run with no further route planning being necessary.

Starting positions for all of the actors are either chosen randomly or loaded from stored values; positions are checked to ensure that actors will not start in situations where they are so close to others that a collision would rapidly occur.

The map of the scene is used throughout the route planning process: actors are given full access to this map. This might seem unrealistic – one might expect actors to have to build up their own map of the scene as they walk through it. However, it is a good simulation of people walking through a building or area that they are familiar with, such as the high street in their town, or their office building; in these situations people will have a very good idea of the route that they should take. It is therefore simply the job of the route planning algorithm to determine this 'familiar route' for each actor. (This idea is discussed further in section 8.2.3.)

5.4.2.1 Scene design

In this system a scene is composed solely of areas of pavement, walls and roads, allowing a simple street scene to be constructed. The interior of a building can be re-created using walls, with gaps representing doorways: this allows an evacuation to be simulated.

Clearly the ability to add more complicated features to a scene would make the simulation more visually attractive. However, most objects can be represented by, say, a wall, as the effect upon the behaviour of the actor will be the same. Given that the aims of this system are to test collision avoidance behaviour, rather than to produce a photo-realistic animation, this approach is sufficient.

5.4.2.2 Map

The map of the scene is simply a three-dimensional array, with each entry in the array representing a map square at a certain height; the value of each entry represents the type of square, such as pavement, wall, or road. Routines exist to convert between world coordinates and map square coordinates.

The advantages of this approach are:

- Easy to create: a simple map editor can be used to 'colour in' different squares.
- Simplifies route planning: the route only needs to be planned at the granularity of squares, not any lower, as obstacles must take up a whole square.
- Compact: the amount of data needed to represent the map is small compared to a system where exact positions for objects are held in vector coordinates.

However, it has the following limitations:

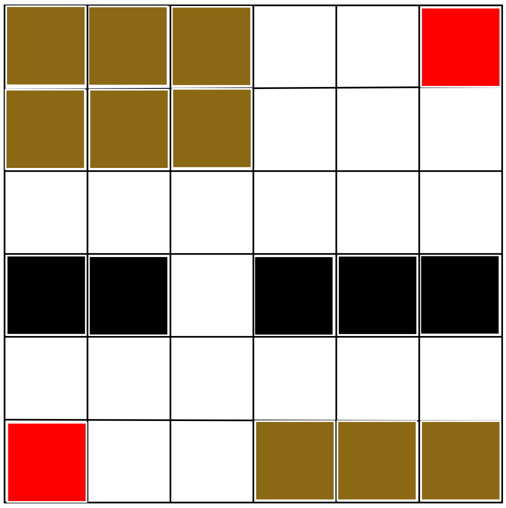

Figure 5.16: An example of a simple map

- Objects can only be sized in multiples of squares.
- Actors are always given routes passing through the centres of squares; this means they may detour around an obstacle more than is necessary.

For this system it was decided that the advantages of the map-square approach outweighed the limitations: the environments to be created are simple, consisting solely of walls and roads, and so exact accuracy in placement is not necessary; if it were to be, the size of map squares could be decreased to allow finer control. More fundamentally, the goal of this work is not to create an original route planning system, just to have a simple system to give the actors something to do.

An example of one layer of a simple map is shown in figure 5.16, with white squares representing pavement, brown for buildings, black for a road (with a pedestrian crossing), and red for miscellaneous unwalkable areas.

5.4.2.3 Choosing goals

Goals are chosen by selecting a random coordinate within the confines of the map. That coordinate is converted into a map square location; if that square is a 'pavement' square then the goal is kept, else another is chosen. In addition, the goal must not be in the same map square as the previous goal – this would cause unnatural short movements in different directions.

Clearly real people's goals are far more complicated than this: they will want to walk towards particular buildings, such as shops, rather than to a random piece of pavement. However, this system is sufficient for the purpose of giving the actors simple tasks to perform.

5.4.2.4 Route planning

The A* search algorithm, originally described by Hart, Nilsson and Raphael [HNR68], is used for route planning. A brief outline is given here.

The algorithm is initialised with the start and end map squares, and the map to be used. A *node* representing the start square is created: nodes contain map coordinates, an *id*, a link to their parent node, and three variables: f, g and h. g is the 'cost' to get to this node from the start node, h is the 'Manhattan' (straight line) distance to the goal, and f is $g + h$. The starting node has no parent, and no cost to get to; its h value is the straight-line distance between the start and end map squares, i.e. the difference in x coordinates plus the difference in y coordinates. The starting node is added to the *open list* of nodes still to be processed.

The main loop of the algorithm then proceeds:

1. Find the node with the smallest f value in the open list. This is an $O(n)$ operation, looking through all the nodes in turn. If more than one node shares the same, smallest, f value one is selected *at random* – this ensures that actors will choose different, equally suitable, routes.

2. Remove the chosen node from the open list, and add it to the *closed list* of nodes that have been processed.

3. Look at the eight squares surrounding the chosen node's square; ignore them if they are off the map, unwalkable (they contain an obstacle), already on the closed list, or consist of a diagonal walk over part of an obstacle square.

4. For each one, if it is not on the open list, add it. Its g value is set to the g value of the current node plus 10 if this is a horizontal or vertical movement, 14 if it is diagonal (14 being an approximation to the hypotenuse of a right-angled triangle with the other sides each 10 units long); the h value is found as above. Its parent is set to be the current node.

5. It the square was already on the open list, see if its stored g value is higher than the value that would be calculated from the current node. If it is, the current route is shorter than a previously used route. The square's entry on the open list is updated with the smaller g value, and the parent set to the current node.

6. Repeat by looping back to step 1, until a square is considered that has an h value of zero – this is the end square.

An example is shown in figure 5.17. In stage 1, the chosen node is the starting node; f, g and h values are calculated for all the squares around it, and parent links are shown with arrows. In stage 2, the square with the smallest f value has been chosen (shown by the red square). However, all the squares surrounding it are either unwalkable, or would have larger g values if they were changed to come through this square – shown in red and in square brackets. In stage 3, the square with the next smallest f value has been chosen, and f, g and h values have been added to the remaining four of its surrounding squares. The next square to be processed will be the one with an f value of 44, and it is clear that the arrows are forming into the shortest route between the start and the end square.

It is now simply a matter of working backwards from the end square through the parent links, back to the start square, writing the coordinates of all the squares into the actor's route. Each square represents a sub-goal for the actor.

This technique has the advantages that:

- It is guaranteed to find a shortest path between start and end squares.
- It will find different shortest paths each time it is run (if possible), producing more variability.

These are both characteristics of human route-planning: people will not on the whole choose to walk further than they need to, but they will choose different ways of getting to the same place.

The only disadvantage of this technique is that it is moderately slow, principally due to the check for the smallest member of the open list, and the check to see if squares are already in the open list. However, this is not a problem, as it is only usually called in a pre-simulation processing step.

5.4.2.5 Route following

Initialisation At the start of the simulation, after route planning has been carried out, each actor will have a starting position, a goal that it should walk towards, and a route which will lead to that goal. Its velocity is set to be in the direction of the first sub-goal along that route, with its speed set to that actor's default value (section 5.4.1).

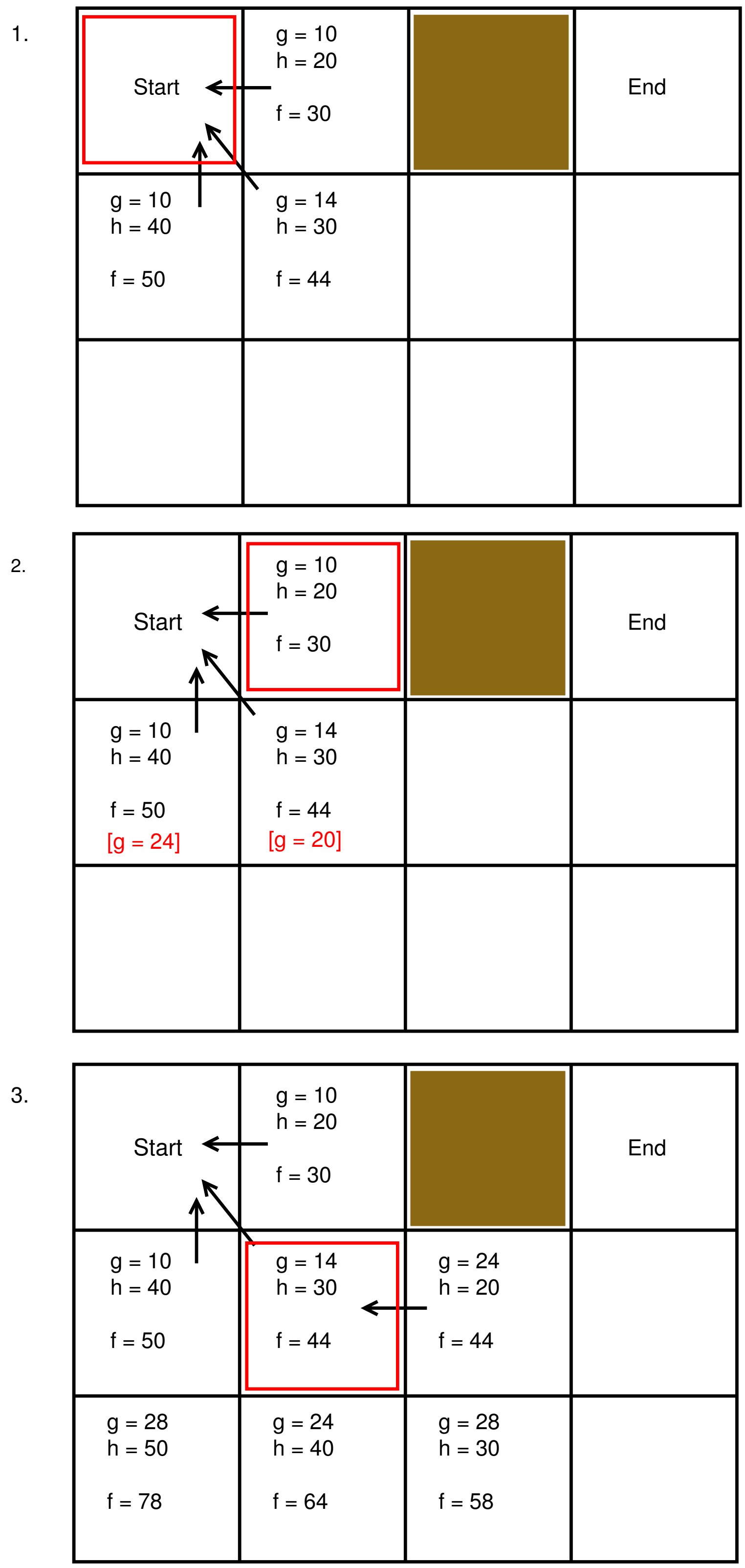

Figure 5.17: An example of A* search route finding

Arrival at sub-goal On every frame of the simulation the position of each actor is compared to its current sub-goal; if it is within half a metre it is considered to have arrived. Exact positions are not required as this would look unrealistic: people are almost always trying to get to a certain area, rather than a particular spot on the pavement.

If the current sub-goal is not the end of the route, the direction of the actor's velocity is altered to point at the next sub-goal along the route. If it is the end of the route then the actor has arrived at its goal; the next goal from the set of previously calculated routes is chosen, and the process is repeated.

5.4.2.6 On-the-fly route planning

There are occasions when the routes planned before the simulation started are not sufficient, in particular when an actor needs to plan a new route to its current goal as it has walked too far off the planned route (see section 5.3.3), but also if an actor has used up its original ten goals.

The same route planning algorithm used in the original processing phase (section 5.4.2.4) is used. However, it is now called in a thread, so that it can be run in the background while the simulation is ongoing.

Each actor is allocated a route planning thread when the system is started, which is normally *asleep* so that it does not take up any processing resources. When route planning needs to be done, the thread is woken up, performs the calculations, and goes to sleep again. This is more efficient than creating a new thread each time planning takes place, as it takes a certain time to perform the creation. Each actor needs its own thread as there is no way of predicting how many actors may need to be performing route planning at the same time.

While the route planning is going on the actor stands still, as a person might stand to think about what to do next.

5.5 Summary

This chapter discussed all of the ways in which psychology research has been used to build the behaviour system for each actor, covering collision prediction, different forms of collision avoidance, and the behaviour of actors after a collision has been avoided, as well as what happens when collision avoidance fails. It also briefly outlined actors' route-planning capabilities, as a way of giving each actor a set of definite tasks to perform. The further ways that psychology research has been used to vary the behaviour of actors by giving them different sets of emotions will be discussed in chapter 7.

All of the rules of behaviour covered in this chapter will be evaluated and discussed in chapter 6.

Chapter 6

Implementation and Evaluation

The previous chapters described the algorithms that have been designed to produce a simulation of vision, attention and behaviour. In order to evaluate these algorithms a system to visualise the results was needed. This chapter will start by describing the 3D simulation system created, and move on to an evaluation of all the aspects of behaviour, presenting examples from the simulation.

6.1 3D Simulation

The decision was made to produce a 3D simulation of the actors and their surroundings, rather than a more simple 2D system. The advantages and limitations of this approach are now discussed.

- The main advantage of using 3D is realism: while the goal of this work is not to produce a photo-realistic simulation, it is desirable to produce results which look vaguely realistic, as this makes it easier for observers to judge whether the behaviour is accurate; this would be much harder on a simple top-down 2D system, perhaps only involving moving circles. Part of the evaluation will involve user assessment of performance, so this is an important concern.

- Using a fully 3D system enables the user to observe the action from any angle; this is useful for testing purposes, so that the user can see exactly what is happening during a collision.

- Conceptually, it should be easier to create a 2D system than a 3D one. However, a brief examination of the options available in Windows for displaying 2D graphics, such as DirectDraw, reveal that they are at least as complicated to construct as their 3D counterparts.
- One drawback to a 3D system is that performance will be worse than in 2D, due to the overheads of drawing 3D models of all the actors. However, this concern is outweighed by the benefits.

Having previously experimented with OpenGL and found the architecture to be non-intuitive, I decided to use DirectX[1] for this work. For ease of use I elected to use C♯ in Microsoft Visual Studio[2], and Managed DirectX. This version of DirectX allows more of the technicalities of window creation and rendering setup to be handled automatically by the library, with very little loss of performance.

6.1.1 Drawing the scene

There are two main aspects to drawing the 3D scene: drawing the actors, and drawing their surroundings.

6.1.1.1 Drawing actors

A simple 3D model of an actor was created; the model is not particularly accurate, but it looks sufficiently like a person for the purposes of this simulation. This was found to be easier than making use of an existing model: the selection of models found online tended to be excessively complicated; this would not only have had an effect upon rendering performance, but would also have made them difficult to animate, particularly for the step-and-slide. By contrast, the simple model created for this system renders quickly, and only required a small number of vertices to be adjusted to produce a step-and-slide animation.

The model was animated using Character Studio (part of Autodesk's 3D Studio Max[3]) to produce a simple walk cycle consisting of 20 frames of animation. In addition, a frame of the actor standing still, and animations of the actor performing the step-and-slide in both left and right variants were created. All of these animations were saved in the MD2 file format, created for the game Quake 2[4]. Originally the plan was to use DirectX's own X format, but it was

[1]www.microsoft.com/directx/
[2]msdn.microsoft.com/vstudio/
[3]www.autodesk.com/3dsmax/
[4]www.idsoftware.com/games/quake/quake2/

found that there was only support in the libraries for non-animated models. The MD2 format was chosen as it consists simply of a copy of the model in each pose of the animation, stored as a list of triangles, making the files easy to parse. The advantage of using the MD2 format over a non-standard format is that it should be possible to use MD2 files created by other people to create a better-looking simulation. (In practice this is not the case, as nearly all third-party MD2 files contain a run-cycle, but no walk-cycle, as this is suitable for their intended use in Quake 2.)

The MD2 file is loaded and processed before the simulation starts. As each frame is read the triangles in it are saved into a *vertex buffer*; normally, this would be done each frame, but as the model does not change it makes the rendering process much faster to perform this step only once. The processing step also reads the text associated with each frame, which is used to separate the different animation cycles (walking, standing and step-and-slide); the start and end frames of each cycle are stored, so that the renderer can automatically loop the cycles.

During the simulation the command to render each actor is issued on every frame of the simulation. Because there are a limited number of frames in the walk cycle it is up to the renderer to decide when to move to the next frame; this decision is based on the current velocity of the actor: an actor that is walking quickly needs to progress through its walk cycle faster so that it continues to look realistic. The renderer will also switch to a different animation cycle (such as step-and-slide) when requested by the simulation.

In addition to drawing the actual actors the renderer can also draw lines above each actor representing velocity and gaze; these make it easier for the user to see what is happening while testing the system. The gaze line is also the only indication of where an actor is looking, as its eyes are not animated in this model – they would be too small to see in anything other than close-ups, and the normal view of the simulation is top down and zoomed out to show a large area.

6.1.1.2 Drawing surroundings

The layout of walls, roads and pavement is determined by the map of the scene. The content of each map square is examined, and the appropriate object drawn; for example, a wall is represented by a vertically-stretched cube, covered in a brick texture. Because most of the scene will normally be covered in pavement, initially a large concrete-textured polygon is drawn across the entire world; this will be hidden by any other object that is in a square.

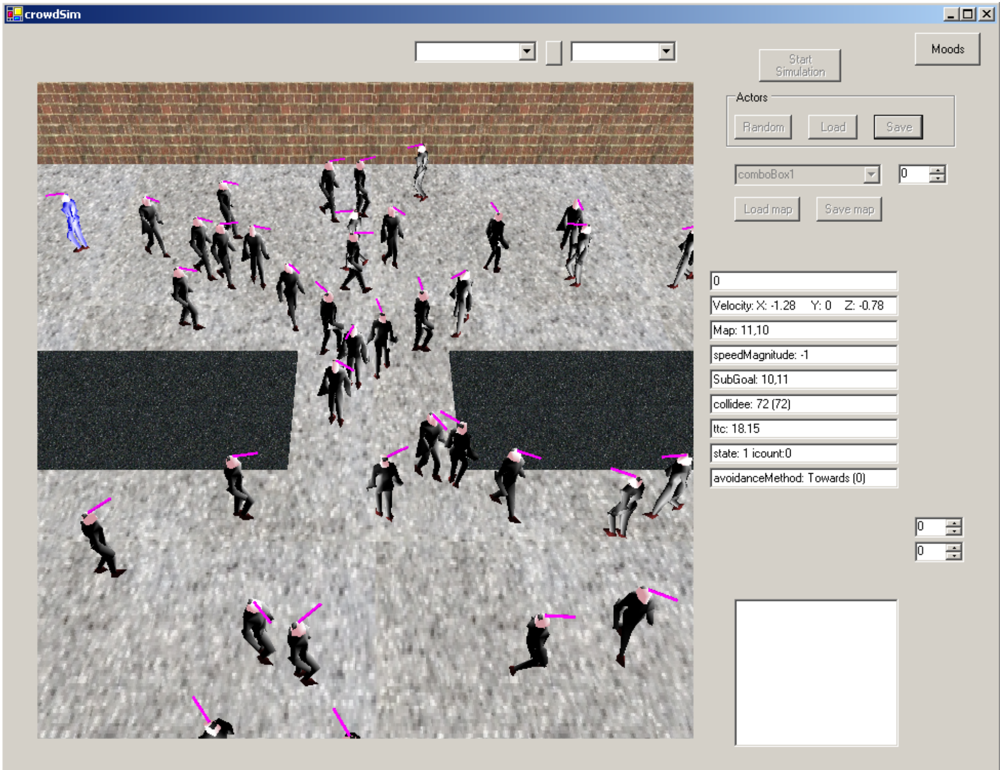

Figure 6.1: An example of the graphical simulation, along with the textual information presented

6.1.2 Presenting information

While the system is being tested, it is helpful if information about the current state of the actors can be displayed, so that the user can understand what is happening. This would also be useful in a final system when a new simulation is being set up. Relevant information is displayed both textually and graphically.

6.1.2.1 Textual information

Information is displayed on screen for the currently selected actor. This includes:

- velocity and velocity magnitude;
- current map square and sub-goal in map co-ordinates;
- collidee for the current collision, and time to collision;
- what state the actor is in (whether it is avoiding a collision, whether it is stuck), and what collision avoidance technique it is using.

An example is shown in figure 6.1.

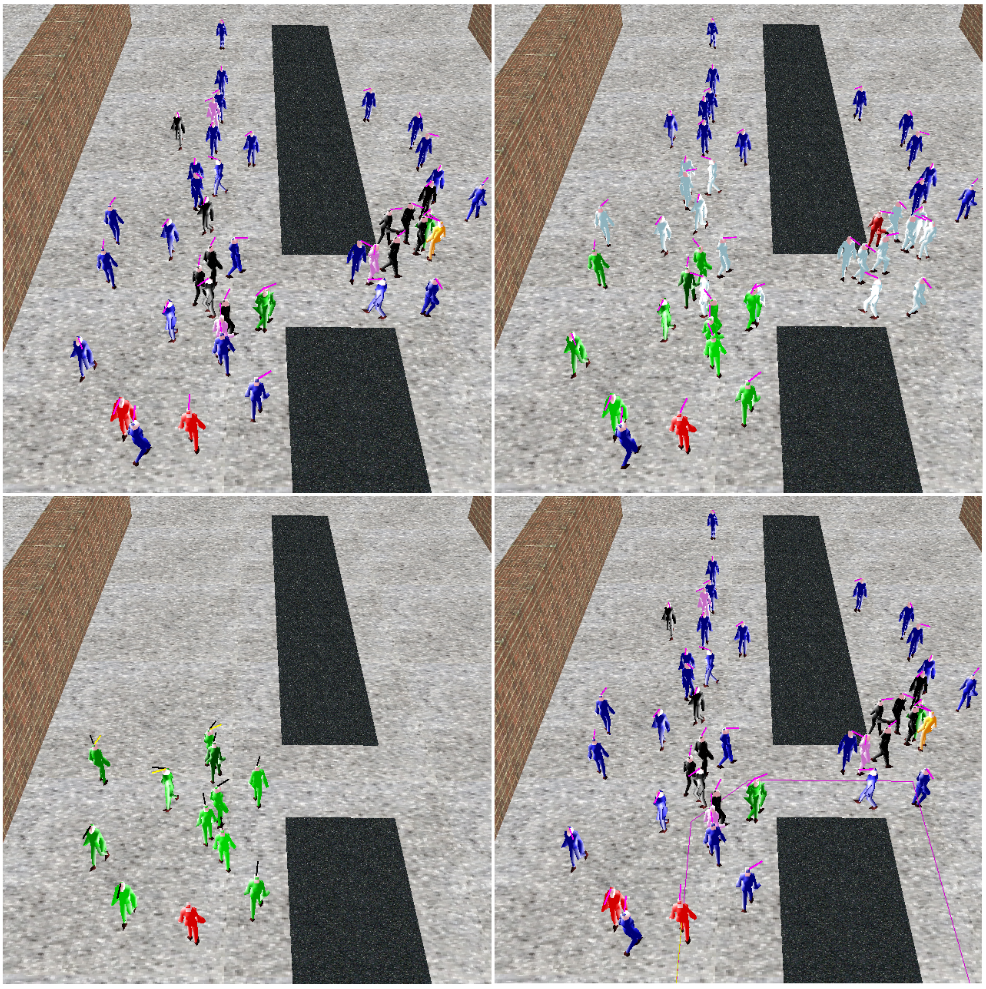

Figure 6.2: Four different options for graphical display of information

6.1.2.2 Graphical information

A large variety of information can be represented, usually using different colours for actors, as demonstrated in figure 6.2:

- Top left: In the standard mode actors are coloured based on what they are doing currently: avoiding a Towards collision (black), slowing down to avoid an Away collision (violet), overtaking (turquoise), avoiding a glancing collision (gold or black), in deadlock with another actor (white), or side-stepping to get out of deadlock (green). Selected actors (section 6.1.3.1) are highlighted (red).
- Top right: Another mode colours actors based on what the most recently selected actor knows about them: there are colours for actors that are

in its peripheral vision queue (yellow), spontaneous looking queue (light blue), and explicit request queues (purple). If an actor is being monitored then it is coloured based on the time since it was last looked at – bright green for an actor just looked at, down to black for an actor not seen for more than 40 frames. This makes it very easy to see where the subject is looking at any point, as actors will change colour as they are looked at and change state; it also shows which actors the subject is monitoring frequently, as they will stay a fairly bright green, whereas less important actors will more often be fading to a dark green.

- Bottom left: Actors can be displayed in the positions stored in the most recently selected actor's mental model – with actors again coloured based on which attention queue they are in. Actors which the subject is not aware of will not be displayed. In this mode one of the actors can be marked, and it will remain marked in the standard visualisation mode: this enables the user to compare the actual and mental model positions of other actors.

- Bottom right: The route that the most recently selected actor has planned can be displayed as a purple line between its starting position and its goal; simultaneously the route that the actor has actually taken is displayed in yellow – this grows behind the actor as it walks. This enables the user to see how closely the actor has kept to its planned route, and how often it has had to divert from it.

6.1.3 Interaction

While the simulation is running the user is able to interact with it in a number of ways: selecting actors, controlling the flow of time and selecting a viewpoint.

6.1.3.1 Selecting actors

The user is able to click on the 3D display, and the system will automatically select the actor nearest to that location; this works no matter what viewing angle is currently in use. The technique for this calculation is given in appendix A.4. By moving the mouse around, the user is able to select multiple actors; the procedure is called each time the mouse moves.

6.1.3.2 Controlling time

When new ideas for a simulation are being tested, it is helpful if the user is able to control the flow of time: to pause, replay or see events in slow-motion. In particular, the ability to replay a period of time means that a situation can be observed from a different angle.

While the simulation is running normally, information about the current positions, velocities, states, and other relevant attributes of all the actors is recorded. The number of frames of information to collect is configurable, but defaults to 500 so as not to take up excessive amounts of memory.

At any point the user can pause the simulation and rewind through the stored information. The information can be played back forwards or in reverse, and at a number of different speeds, or stepped through frame by frame. This allows the user to see exactly on which frame a certain behaviour started, and if that was the correct decision. During playback the transition between the end of the stored information and the ensuing live simulation is seamless.

6.1.3.3 Selecting viewpoint

The user is able to view the simulation from any angle; this is not only useful for testing, but for producing an interesting final result. The virtual camera can be moved in any direction (forwards and backwards, sideways, up and down) using the mouse; the viewing direction can be panned and tilted.

The camera can also be set to track the currently selected actor: the camera will stay fixed, but the actor will always be centred on the screen. Alternatively the camera can be set just behind, in front of, or above the tracked actor, and will move with it.

6.2 Results

This section presents graphical evidence of the techniques described in the last two chapters. Firstly some of the basic behaviour rules are demonstrated, followed by an examination of the effects of attention simulation. Finally, some of the more advanced effects achieved by combining the simple behaviour rules, along with attention simulation, are presented.

Following this, quantitative results of the effects of attention simulation are given and analysed, along with a brief discussion of performance.

6.2.1 Graphical evidence

Figure 6.1 showed an example of the type of simulation that can be created by this system: a street scene consisting of pavements, a road with a pedestrian crossing, and walls (representing buildings); an animated version of this scene

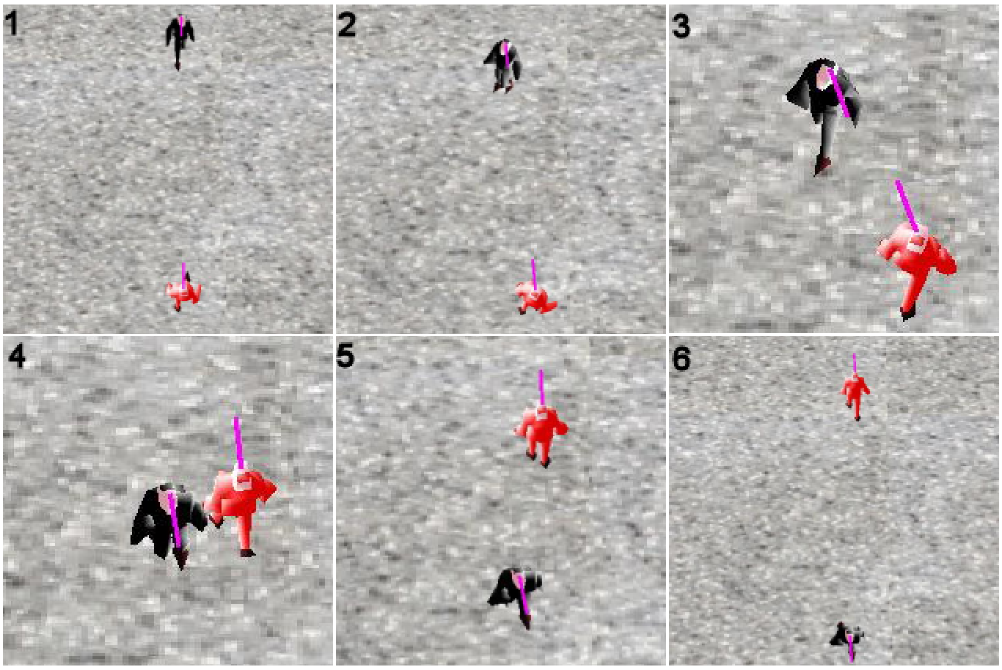

Figure 6.3: Avoiding a Towards collision by changing direction

is shown in video 6.1[5]. The various different techniques which went into the making of this type of simulation are now covered in turn.

6.2.1.1 Simple behaviour rules

The simulation of behaviour described in chapter 5 relies on taking simple rules derived from psychology research, and combining them to produce a complex system. At the heart of these rules are the varying ways of avoiding collisions; these are now demonstrated.

Towards collisions As described in section 5.2.3, a Towards collision can be avoided either by a change of direction, or speed, or both. In figure 6.3 (video 6.2), the red actor is to the right of the black actor; therefore, according to psychology research, it should avoid on the right-hand side, with the smallest detour required to avoid an actual collision. The actors start avoiding the collision as soon as the simulation is started to minimise the size of the angular direction change needed. After the encounter, both actors can be seen to return to their original path, as desired.

An example of an actor changing speed to avoid a collision is shown in figure 6.4 (video 6.3). The single actor coming from the bottom can easily avoid the

[5] See enclosed DVD for video files, in XviD, WMV and Quicktime formats.

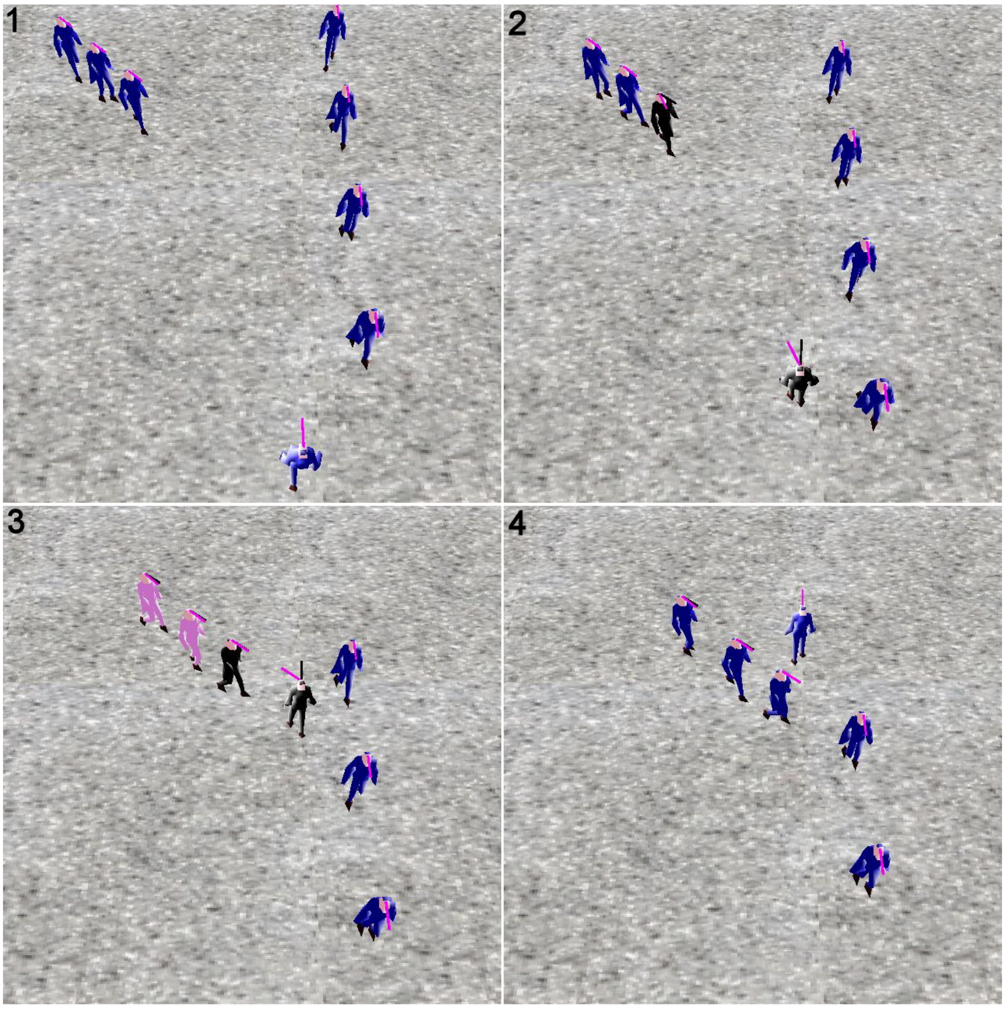

Figure 6.4: Avoiding a Towards collision by changing speed

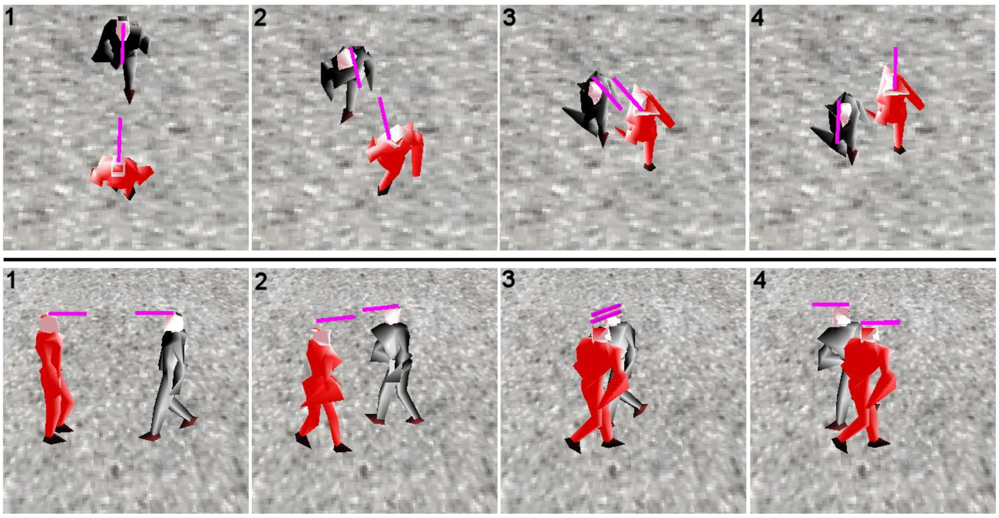

Figure 6.5: The step-and-slide manoeuvre seen from the top, and from the side

three actors coming towards it from the left by increasing its speed by 30%, to get past the collision point before the other actors get there. (The presence of the column of actors on its right means that the single actor cannot avoid the collision by moving right.) The lead actor of the group of three slows down by 50% as well, which, incidentally, means the two actors behind it have to slow down (chosen here) or overtake. In these diagrams the actors are coloured in blue normally, and black when they are changing speed to avoid a Towards collision; violet is used for slowing down to avoid an Away collision.

Figure 6.5 (video 6.4) shows the alternative way of avoiding a Towards collision, if normal avoidance behaviour has failed: the step-and-slide manoeuvre (section 5.2.3.4). For this demonstration the actors have deliberately been placed so close together that they are unable to avoid the collision by changing direction or speed. The top row of pictures shows a top-down view; the bottom row shows the same points in time from a side-on perspective. This type of situation can occur either as a result of a collision only being predicted at the last moment, due to attention simulation, or due to sudden changes of velocity, causing a collision which would not otherwise have been expected.

Away collisions An Away collision can be avoided either by slowing down, or by attempting to overtake, as described in section 5.2.4. Evidence of the effects of slowing down will be shown in section 6.2.1.3. The overtaking procedure is shown in figure 6.6 (video 6.5).

- Shot 1: The red actor approaches three other actors; as it is walking faster than them, it predicts a collision with the central actor, and chooses to

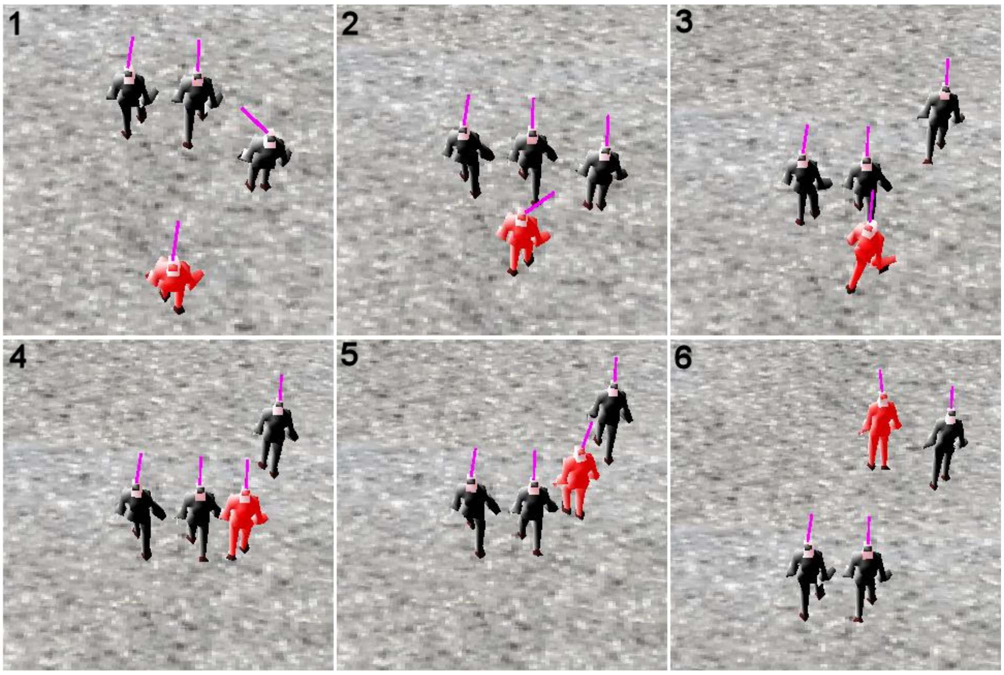

Figure 6.6: The process of overtaking

attempt to overtake it.

- Shot 2: It slows down as there is no space to overtake: it cannot overtake on either the left or right side of the central actor, due to the two flanking actors (and it will not overtake more than one actor at the same time). It will continue to walk at the same speed as the central actor, never getting too close to it, as doing so would make it harder to overtake, and be anti-social; it also walks slightly to one side of the actor, as desired.

- Shot 3: At this point the right-hand actor of the three, which is walking faster than the others, has left a sufficiently large gap that there is now room for the red actor to overtake. It therefore increases its speed and sets off around the right-hand side of its collidee.

- Shot 4: The overtaking procedure is still ongoing: even though the red actor is now just past its collidee, it continues walking (at its increased speed), as cutting back in now would cause a side-on collision.

- Shot 5: The danger of collision is now over, so the actor is free to return to its original path, and slow down to its normal speed.

- Shot 6: The actor is now back on its original path.

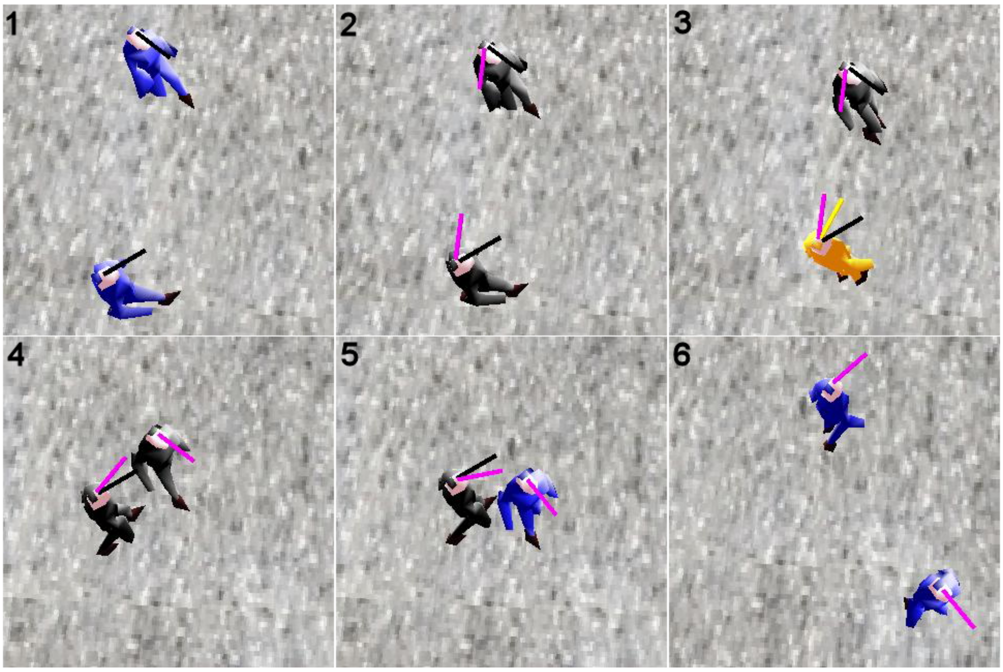

Figure 6.7: Avoiding a Glancing collision

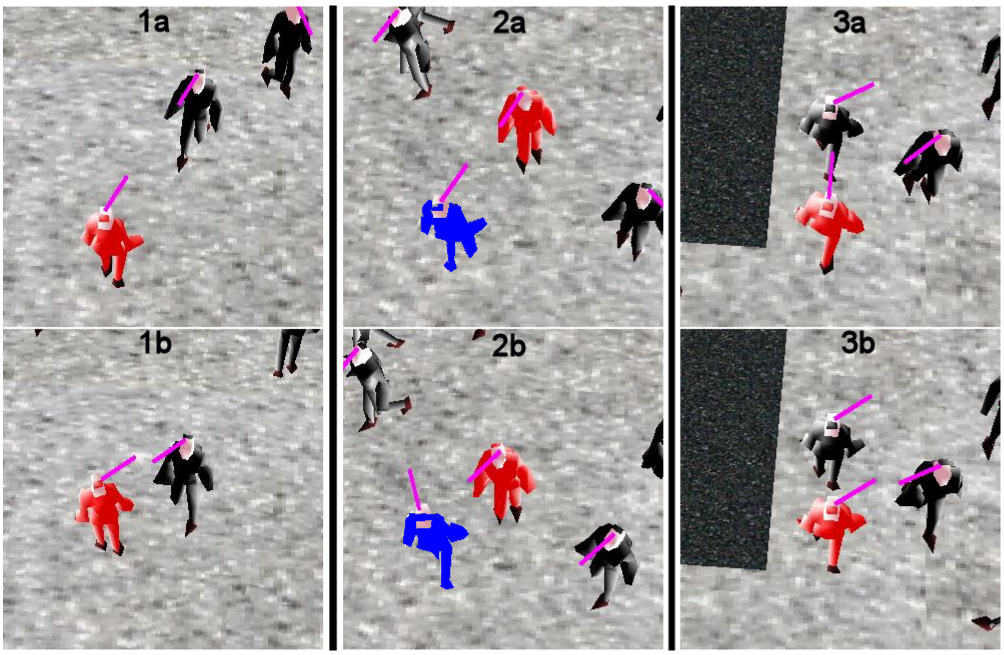

Figure 6.8: Three examples of an actor's gaze movements

Glancing collisions Glancing collisions are usually avoided by changing speed, sometimes combined with a small change of direction. In figure 6.7 (video 6.6) the two actors are on course for a glancing collision. To avoid this, the lower actor slows down by 50% (in shot 2) and makes a minor change of direction (in shot 3); once it has made this decision, there is now no need for the top actor to make any changes, and it can proceed at its normal speed. After the collision has been avoided, the first actor returns to its original speed and direction (shot 6).

6.2.1.2 Attention simulation

Once the basics of behaviour simulation were in place, the attention system was enabled, to determine what effect it would have upon behaviour. A few of these effects are now presented.

Gaze movement As it would be impossible to see the eye movements of individual actors, the direction of their gaze is shown in these figures and videos as a purple line above an actor's head. Figure 6.8 shows three examples of an actor's gaze movements; in each case the a and b shots are just a few frames apart. Shots 1a and 1b show two actors both monitoring each other as they perform collision avoidance, to ensure that neither is changing course (which might necessitate a rapid move to compensate). Shots 2a and 2b show that actors do not look at each other all the time during collision avoidance: it is still important to check on the positions of others, as the blue actor is doing. Finally, shots 3a and 3b show the red actor trying to overtake the actor in front of it: first it looks at that actor, then it looks to the side at the actor which has been blocking its overtaking path. Video 6.7a shows animated versions of these three examples, and video 6.7b shows one actor's gaze movements as it walks through the scene.

Ignoring others Section 4.3.5.2 discussed some of the ways that an actor's mental model can be incomplete or inaccurate; one of these was that an actor may be concentrating so hard on actors for which it has predicted a collision that it does not notice other nearby actors. An example of this situation is shown in figure 6.9 (video 6.8).

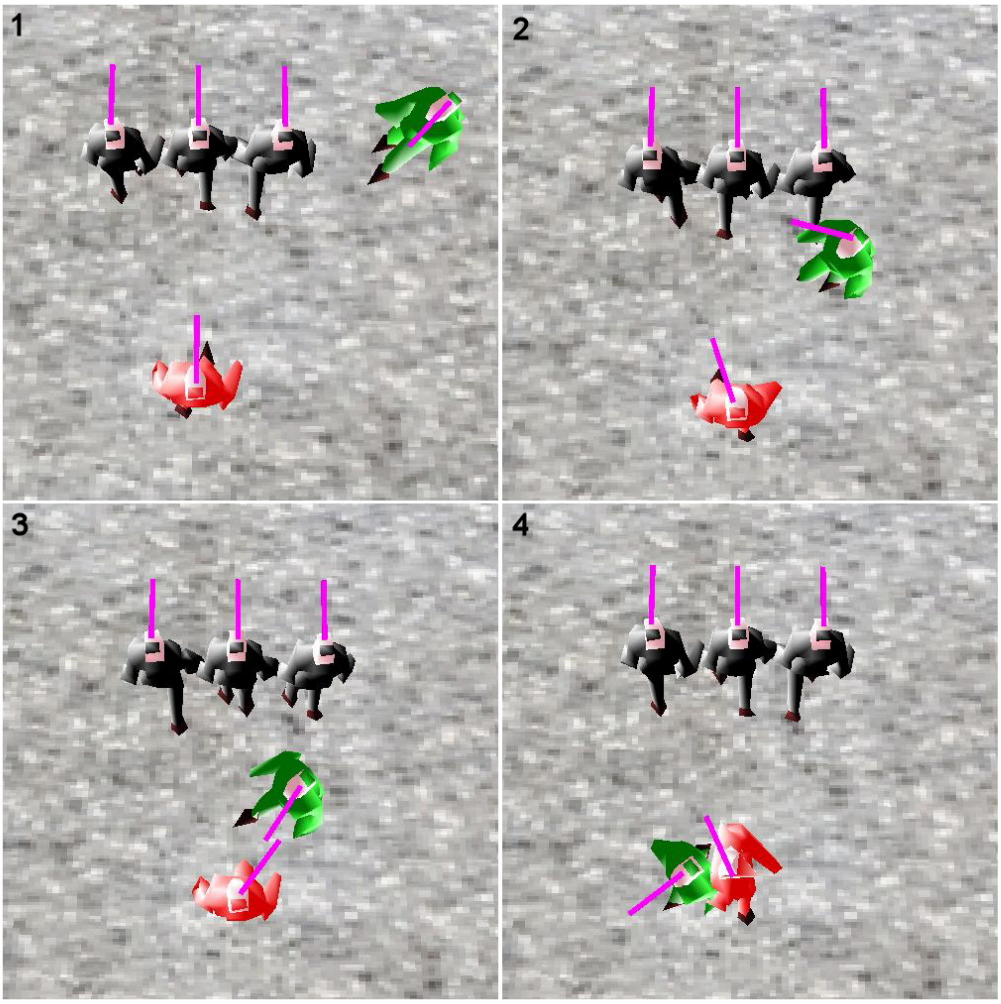

Figure 6.9: A situation where inaccuracies in an actor's mental model can arise

- Frame 1: the red actor is looking at the three black actors in front of it. It has been walking faster than them, and would like to overtake; currently it is unable to do so, as the three actors together are creating a blockage. Therefore, it has slowed down behind them, to avoid an actual collision, but is closely monitoring them so that it can overtake as soon as a gap emerges. The green actor approaches from one side: for the purposes of this illustration, it has been instructed to ignore any collisions it predicts.

- Frame 2: the red actor is still concerned with the actors directly in front of it, in particular the actor on the left of the line of three. It has not yet detected the presence of the green actor: by now, it will have been seen in peripheral vision, and will therefore be scheduled for observation as a priority, but as yet the red actor will not consider it a threat.

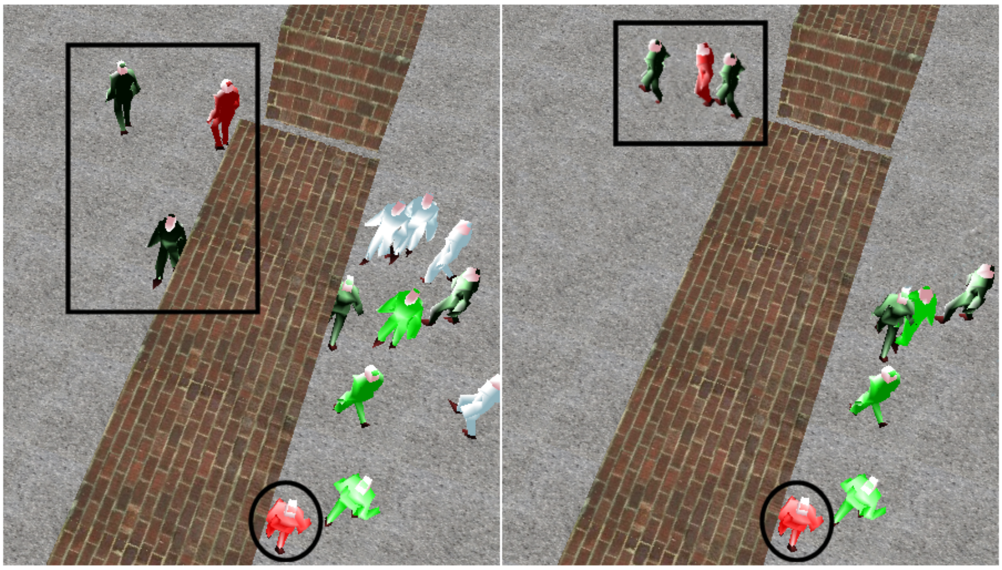

Figure 6.10: Left: real positions of actors. Right: positions of actors according to mental model of circled actor

- Frame 3: the red actor finally looks at the green actor.
- Frame 4: the knowledge about the position and velocity of the green actor came too close to the collision for any action other than the step-and-slide to be taken, and a partial collision occurs.

Clearly in normal operation the green actor would have detected the pre-occupied red actor, and made its own efforts to avoid it. However, the situation shown here is not unheard of in the real world: some people are so self-absorbed that they will not notice a collision is imminent, or so rude that they will not try and avoid collisions, expecting others to get out of their way (see section 7.3.2.5).

Mental model As described in section 6.1.2.2, the mental model of any actor can be displayed visually, with the positions and velocities of all the actors shown according to those values from the mental model.

An example of this is shown in figure 6.10, with the real positions on the left, and the mental model of the circled red actor on the right. It can be seen that the mental model positions of the actors on the same side of the wall as the red actor are fairly accurate, although a few of the further away actors have not yet been observed.

However, the positions of the actors on the other side of the wall, marked here with a rectangle, are significantly inaccurate. This is because they have

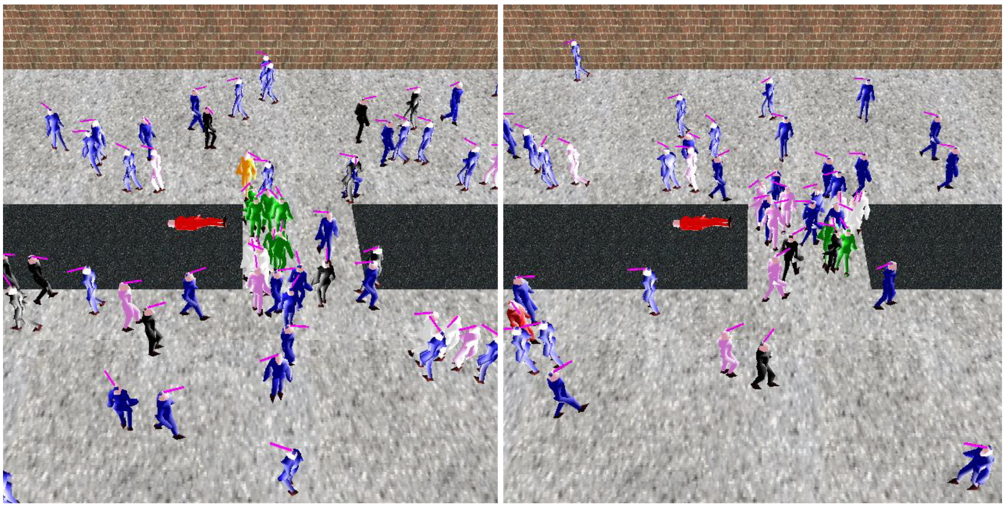

Figure 6.11: The effect of a distraction – in this case an actor lying down on the road

passed through a gap in the wall since the last observation; they are therefore out of sight of the red actor, and so it can only base its estimate of their current location on the positions and velocities it saw last – which would have been when they were walking straight towards the gap in the wall. It assumes they have continued to walk in this direction, whereas in fact one of the actors turned left after walking through the gap. (As discussed in section 4.3.5.2, this is not a problem, but a feature of attention simulation.) These actors will shortly be removed from the mental model, due to being out of sight for too long, which solves the problem of inaccurate information.

Video 6.9 shows a different scene in which the viewpoint alternates between the real view of the scene and the mental model view of the red actor (indicated by the caption "Mental model"). It can be seen that the subject only ever knows about a limited subset of the other actors, principally those right in front of it. However, its estimated positions for those actors are of a high accuracy, as they are being observed frequently due to their importance to the subject. The subject thus normally has a good internal model of the actors with which it is most likely to collide.

Distractions One final example is of the effect that distractions may have on the simulation of attention: if an actor is not looking where it is going then it may lead to more collisions with actors that it is not aware of. A distraction is simulated by giving each actor a series of explicit requests to look at the actor causing the distraction; the actor will continue to monitor its environment occasionally, but the majority of its time will be spent looking at the distraction.

Figure 6.11 (video 6.10) shows two isolated frames from a simulation in which there is a distraction next to the pedestrian crossing: somebody is lying down on the road (perhaps to protest about unemployment among film extras due to the use of virtual actors). As the actors walk over the crossing they notice this distraction and repeatedly look at it – as can be seen from the purple gaze pointers. In looking at the distraction they have less time to look ahead of them, at actors they might collide with. This leads to a large increase in the number of collisions happening on the crossing, which leads to it almost getting blocked at times. (See section 6.2.2.2 for a quantitative analysis of the situation.)

6.2.1.3 Complicated behaviour

The simple behaviour rules demonstrated in section 6.2.1.1 combine together, along with attention simulation, to produce complicated behaviour patterns. Some examples of this are now presented.

Lane formation An example of lane formation is shown in figure 6.12; four clear lanes of people are highlighted here in red, green, yellow and blue, and many of the other actors have spontaneously paired into groups of two. Lanes form particularly at the crossing: as it is the only way across the road it has a large number of actors all going the same way, and they slow down and follow each other across; given the number of people going each way on the crossing, overtaking would be particularly difficult. Video 6.11 shows more examples of lane formation – often visible as a group of violet actors (their colour indicating that they have slowed down).

Evacuation One of the primary uses of this system is in the simulation of evacuation from buildings; in particular, testing different designs to determine which allows a group of people to leave in the shortest amount of time.

Figure 6.13 shows a crowd of actors who all have the same goal of passing through the gap in the wall; this is the same situation as people trying to get through a door in a building. Actors approach the gap from all sides, which makes possible collisions of all three types. A number of actual collisions do occur, due to inaccuracies in the mental model; as collisions occur the gap can at times become blocked by groups of people, slowing down the progress of all.

Figure 6.14 shows an alternative layout, with two narrow gaps, each half of the width of the gap in figure 6.13. Even though the total width of the opening

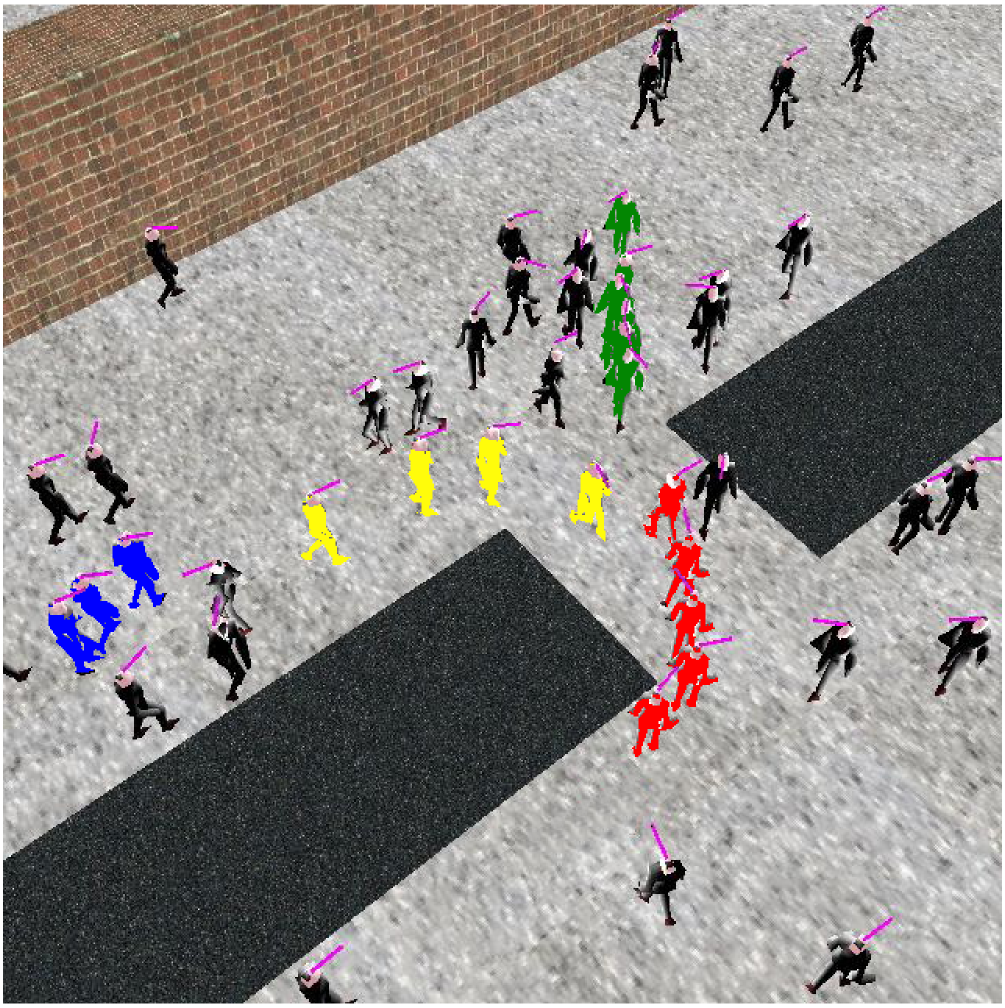

Figure 6.12: Lane formation

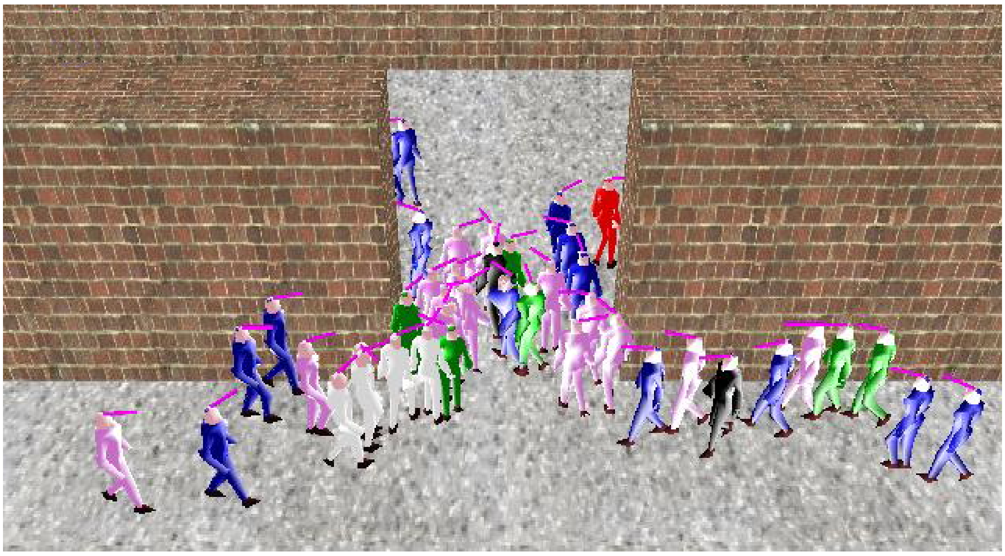

Figure 6.13: Actors evacuating through a wide door

Figure 6.14: Actors evacuating through two narrow doors

is the same in both cases, there are fewer collisions in figure 6.14, and actors can evacuate in a shorter amount of time. (See section 6.2.2.3 for a quantitative analysis, and section 6.3.4 for a discussion of the causes of this effect.)

Both layouts are shown in video 6.12.

6.2.2 Numerical results

As well as testing the system graphically, various types of quantitative analysis were performed, to evaluate the success of the new features of the system. Tests carried out were:

- A comparison of attention simulation with other types of awareness.
- Calculating the effect of actors being distracted.
- Testing different building layouts.

6.2.2.1 Testing attention simulation

Two types of numerical testing were used to compare attention simulation with other types of awareness: counting the number of collisions taking place, and measuring the time taken for a group of actors to evacuate from a building.

Number of collisions It is useful to count the number of collisions that occur during the course of a simulation, as it is a measure of the accuracy of actors' mental models of the scene: the less accurate the model, the more collisions will take place. Further discussion of this point follows in section 6.3.3.1.

Testing was carried out on two scenes: the evacuation situation shown in figure 6.13 and the street scene shown in figure 6.1, to ensure that any effects noticed were not present in just one type of simulation. Each scene was populated with 100 actors. For each scene five different, random, sets of starting positions for the actors were used, to stop results being affected by any special-case configuration of actors. For each of these sets of starting positions the simulation was run ten times; because there are a number of random components that affect the actions of the actors, these runs will all return different results. (This approach was taken, rather than using new starting positions each time, to explore the different outcomes that can result from the same starting positions.) Therefore, there were a total of 50 test situations for each of the two scenes. Each simulation was run for a period of 200 frames (10 seconds).

The occurrence of a collision is detected by the 'Collisions with other actors' function described in section 5.2.6.2, and is added to the current collision count;

each collision will only be counted once, even if it takes a number of frames to be resolved.

The 100 tests described above were performed first with each actor having total awareness of those around it. In other words, each actor is able to access exact position and velocity data for all other actors at any time; this is similar to the system used in some previous work. Secondly, the tests were run using the attention simulation and mental model system described in this work (chapter 4). Finally, the tests were repeated with actors having absolutely no awareness of each other; in other words, they will walk straight towards their goals as if they had their eyes shut.

An automated system was created to perform all the tests: it launches the simulation the desired number of times, stops it after a specified time, and records the number of collisions that occurred.

Firstly, comparing total awareness with attention simulation, in the evacuation scene there was an average increase in collisions of **62%** when using attention simulation; in the street scene there was an average increase of **56%**.

Comparing total awareness with no awareness, there was an average increase of **126%** when using no awareness in the evacuation scene, and of **145%** in the street scene: in both scenes nearly all of the actors collide and stop, and cannot resolve the collision as they are unable to see the actor they have collided with.

Further numerical details are given in appendix B.1, and these results are discussed in section 6.3.3.1.

Time to evacuate As was just shown, using attention simulation increases the number of collisions between actors that occur; this will have the effect of slowing down the overall motion of the crowd. Incomplete mental model information will also lead to slower movement, as collisions cannot be avoided so efficiently. (These factors are discussed further in section 6.3.3.2 [Speed of movement]). Therefore it is clear that the use of attention simulation will lead to a change in the time taken for a crowd of actors to evacuate from a building.

Testing was carried out from five different starting positions of actors, with five runs of the simulation for each, following the same logic as above. The scene used was the one shown in figure 6.13, containing 100 actors. An automated system carried out the tests, measuring the number of frames between the start of the simulation and the time when the last actor escapes through the gap in the wall.

For this test, only total awareness and attention simulation could be compared: if the actors were given no awareness there is no way they would ever be able to escape from the building, as they get stuck after their first collision.

With attention simulation turned on, it took an average of **54%** longer for all actors to evacuate compared with total awareness.

Numerical details are given in appendix B.2, and these results are discussed in section 6.3.3.2 [Evacuation times].

6.2.2.2 Effects of distractions

One of the effects which can be modelled using attention simulation is actors being distracted; an example of this was shown graphically in section 6.2.1.2 [Distractions]. The effect can also be measured quantitatively, by counting the number of extra collisions.

This test was carried out by running the simulation both with and without the distraction of the actor next to the crossing, and measuring the number of collisions taking place over a 200 frame (10 second) period. For each setting the test was run 5 times for each of 5 different starting positions of actors.

It was found that with the presence of the distraction there were an average of **26%** more collisions. Numerical details are given in appendix B.3.

6.2.2.3 Testing building layouts

Figures 6.13 and 6.14 showed two different building layouts – one with a wide door, and one with two narrow doors – and the effects that these had on the ease of evacuation. A quantitative test was also carried out, to confirm which layout allowed people to evacuate in the shorter time.

Both layouts were tested 10 times, with different starting positions of actors for each run, counting the number of frames until the last actor had passed through the gap.

It was found that the two-door layout allowed the actors to evacuate on average **17%** quicker than the one-door layout. Converting from frames to seconds, the average time taken to evacuate was 16.5 seconds compared to 20 seconds. Numerical details are given in appendix B.4.

Clearly many other quantitative tests on building layout could be carried out; this example was chosen as it is representative of the options that need to be considered when designing a building. It is also sufficient to demonstrate the type of evacuation simulation that can be carried out: further experimentation was left for users of the system.

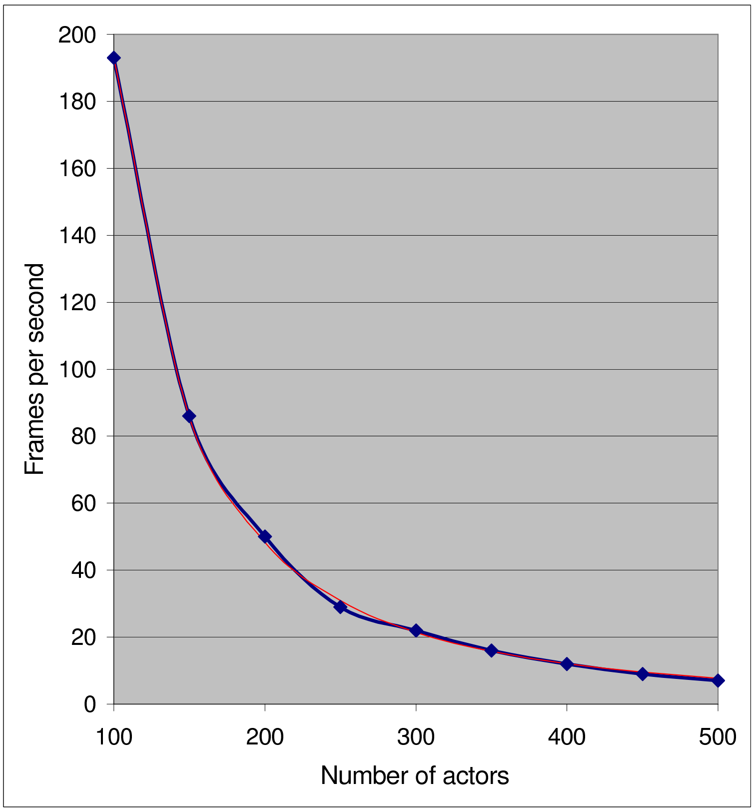

Figure 6.15: Performance for varying numbers of actors

6.2.3 Performance

This system was designed to work in real-time, so that a user would be immediately able to see the effects of his actions upon the behaviour of the actors. Testing was carried out on a moderately high-end PC with an AMD XP2700 processor. As the numbers of actors is increased the frame rate decreases, as shown in figure 6.15 by the dark blue diamonds joined with a dark blue line.

Each actor in the simulation has to consider the position of every other actor in the simulation on every frame, even if only to check that it is too far away to be of interest; due to this, the time complexity is $O(n^2)$. This is also shown in figure 6.15: the red line is proportional to $\frac{1}{n^2}$ – the inverse is used as the time taken to render a frame is inversely proportional to the frame rate.

However, if the crowd is more spread out, then more actors can be discarded at an early stage, as they are too far away to have any effect; therefore, fewer actors have to go through the full processing stages. Figure 6.16 shows how the frame rate increases as the size of the map, measured in grid squares, increases (and therefore as the density of the crowd decreases). The frame-rate appears to be linearly proportional to the size of the map, until the map is large enough

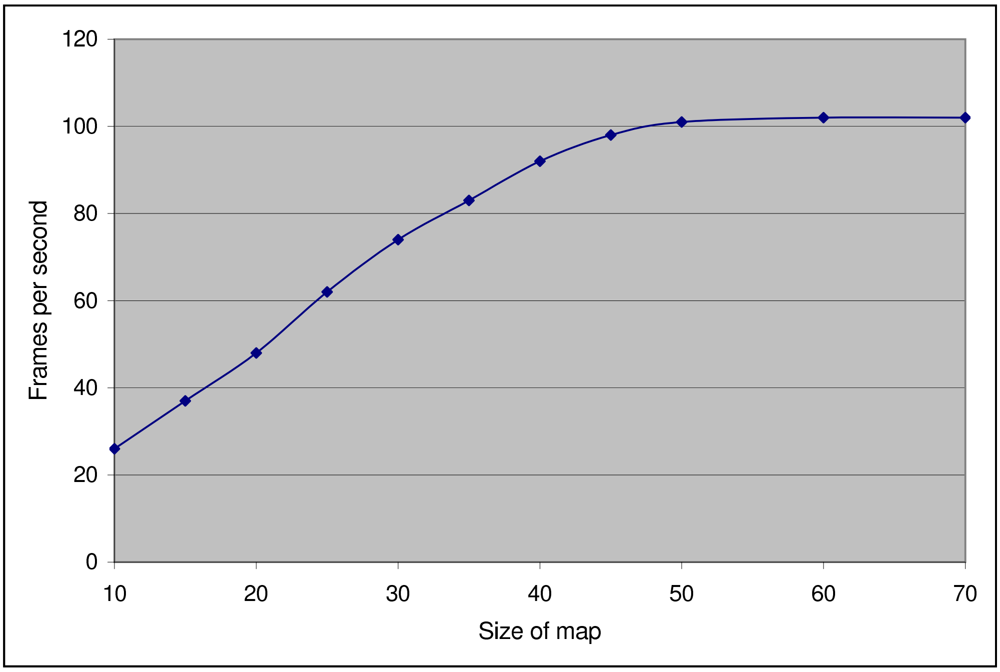

Figure 6.16: Performance for varying size of scene

that actors are often not in range of any others, and therefore do not need to do any full processing.

If very large crowds are required, the system will still function, but no longer in real-time; once a simulation has been calculated, it can be played back at normal speed using the time controls described in section 6.1.3.2. For the size of crowds needed for most purposes, though, the system meets the real-time requirement.

6.2.3.1 Code profile

An investigation was carried out into the division of computational effort between different sections of the code; results are shown in figure 6.17.

The left of the figure shows the division between rendering time and actual behaviour simulation algorithms. It is clear that a lot of time – 45% – is spent rendering actors and their surroundings; this 3D performance was not considered during construction of the system, so it is likely that this could be optimised considerably.

The right of the figure shows the time spent by different components of the behaviour algorithms. It is notable that collision prediction takes a significant proportion of the time – 32% – as it has to consider interactions between many pairs of actors. The ellipse tests (used to check for collisions) are also

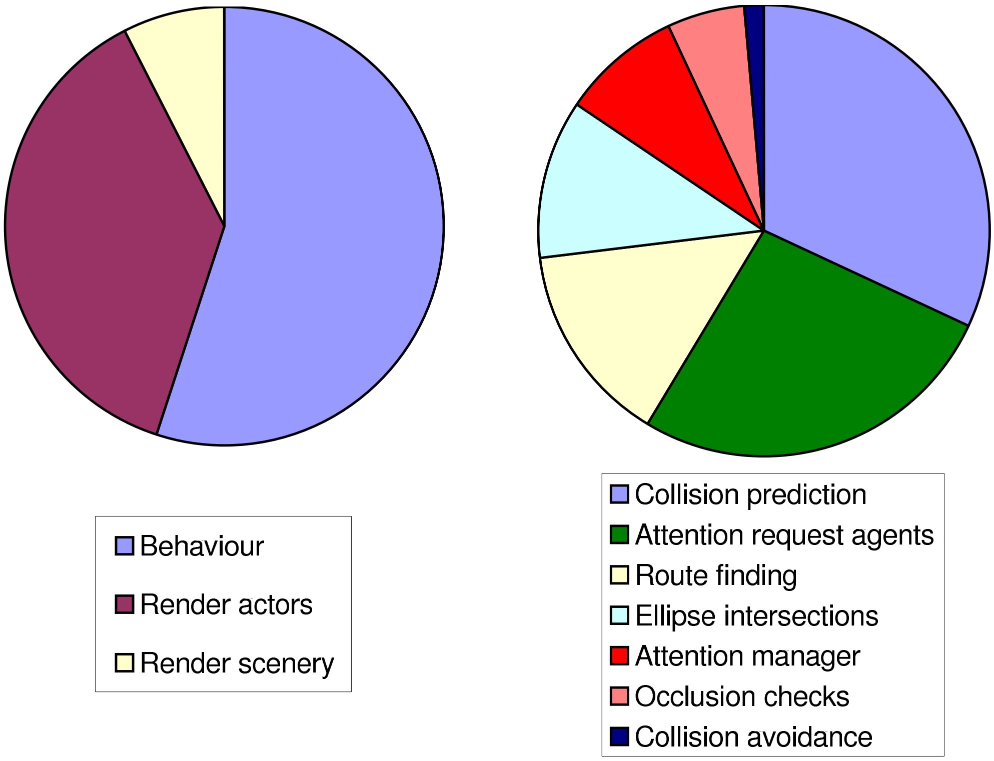

Figure 6.17: Left: Overall code profile. Right: Behaviour code profile

expensive – 11% – despite being simpler than full intersection checks. However, there appears to be no single component which is particularly slowing down the simulation.

6.3 Discussion

Following the presentation of results above, this section discusses and evaluates those results, comparing this system both to previous work and to real human behaviour. The section is divided into four parts:

- improvements to behaviour;
- improvements to eye gaze behaviour;
- effects of attention simulation on behaviour;
- uses of the system.

6.3.1 Improvements to behaviour

This system contains a number of behavioural rules that are either an improvement on previous work, or have not been simulated before, all of which are designed to produce a simulation closely based on real human behaviour.

6.3.1.1 Step-and-slide

The step-and-slide is a pervasive feature of everyday collision avoidance behaviour, described in detail by both Wolff [Wol73] and Collett and Marsh [CM81]. One only has to walk down a moderately busy street to observe the manoeuvre being carried out by other people and, subconsciously, by oneself.

In the face of this evidence it is surprising that no previous work has simulated this behaviour. Instead, the standard way for actors to avoid a collision with someone walking towards them has been for them to make a detour; actors will always make a detour that is sufficient to completely avoid the collision, or if this is not possible then they will collide. However, this behaviour is not an accurate representation of human behaviour, and looks unnatural due to actors being forced to make large changes of direction to avoid each other.

Instead, the step-and-slide manoeuvre simulated here produces a more realistic overall effect. It is used in a number of situations:

- Actors that were late to predict the occurrence of a collision can still sometimes avoid it by using the step-and-slide; without it their only alternatives would be to stop walking, or actually collide.
- Even if it would be possible for an actor to make a suitable detour to avoid a collision, it will not always do so: real people often combine a partial detour with the step-and-slide, so that they do not need to divert much from their course. In particular, an actor will try to avoid making a detour that would require a sudden, large change of direction, caused by an imminent collision.
- Sometimes even if an actor has plenty of time to avoid a collision, it will still use the step-and-slide, particularly if it is in a hurry (section 7.3.2.5).

All of these uses of the step-and-slide lead to noticeably different results compared with solely using detours:

- Actors are able to avoid some collisions that they otherwise would not have been able to.

- Actors will move away from their path less frequently, and by smaller amounts; this will result in actors taking generally straighter paths through a scene, rather than continually changing direction.

- There is the possibility for variation in actors' behaviour, based on the size of detour they choose to make, making the simulation more interesting.

It is for these reasons that the step-and-slide is a significant aspect of human collision avoidance behaviour, and one worthy of simulation. This system is the first to provide that simulation.

6.3.1.2 Passing on correct side

According to the psychology research cited in section 3.1.3.1, there are quite strictly defined rules about which side people will choose to pass their collidee: assuming no other factors are involved they will pass on the shorter side or, if both sides are equal, on the right.

This real-world behaviour has been re-created in this system. When avoiding oncoming ('Towards') collisions, actors will prefer to pass on the shorter side, or on the right if both sides are equal, as long as no other actors block that path. When overtaking others during 'Away' collisions, actors will again prefer to use the shorter side, if possible.

Although it may seem like a minor point, the use of these rules does add to the realism of the simulation by making collision avoidance behaviour look more natural. Even where this is not noticeable, it will at least contribute to the accuracy of the simulation, affecting aspects such as how long it takes to evacuate a building.

6.3.1.3 Overtaking

In previous work, the issue of how to stop actors from colliding with others in front of them has been dealt with in two ways. Either, all actors are set to walk at the same speed, and the situation will never arise. Alternatively, the faster-moving actor will simply move to one side of the actor in front, continuing to walk at the same speed, and will eventually walk past it [TM95].

However, neither of these approaches is a satisfactory approximation of the real behaviour described in section 3.1.4, for two reasons:

- People will not always choose to overtake, preferring instead to slow down.

- People will only overtake if there is space to do so.

Decision to overtake The decision about whether an actor will overtake or slow down will be examined further in section 7.3.2.4, as it is based on the current moods and emotions of the actor. An interim system, giving each actor a single number to represent how much of a hurry it is in, was described in section 5.2.4.

However, giving actors the opportunity to slow down, rather than always overtake the actor in front of them, is a critical aspect of a realistic simulation. A brief observation of a busy street scene will show that many people prefer to slow down, thereby avoiding the sometimes difficult procedure of threading their way through others in order to overtake.

This simulation reproduces this real-world effect, giving a mixture of impatient actors who are continually overtaking, and those that are content to go at the pace of the crowd.

Lane formation The formation of lanes of people was discussed in section 3.1.4.1 as an effect that had been observed by all the quoted psychologists. It is deliberately not programmed explicitly into the system, as it should emerge from a combination of the simple behaviour rules, as it does in the real world: people do not consciously decide to form lanes, they just do so because it is in their best interests.

Lanes form in this system due to the ability of actors to avoid Away collisions by slowing down to the speed of their collidee, rather than trying to overtake. Even if an actor has decided to overtake, it may have to slow down until there is room for it to do so. Therefore a situation will emerge where a proportion of actors will form lanes, all moving at the same speed, while others – who wish to travel faster – will thread their way through, overtaking them.

The fact that lane formation takes place in this system, despite not being explicitly included, is evidence that the simple rules of behaviour given to actors are a good re-creation of the real world. It also demonstrates how simple rules, as long as they are accurate, can combine together to produce realistic complicated behaviour.

Overtaking In previous work, when one actor has decided to overtake another it will begin the procedure without first determining whether there is room for it to be completed; it will simply move to one side and try to pass by. This technique can be problematic: if another actor is blocking the path to the side of the collidee, the actor will have to continue moving sideways. In a dense crowd it may end up moving from side to side, trying to find a way through. In addition, if it meets an oncoming actor during overtaking, that actor will have to be avoided as well; if there is no space to do so, the actor may become trapped. Both of these behaviours can seriously detract from the realism of the simulation.

Therefore a novel feature of this system is that once an actor has decided to overtake it will confirm whether there is actually space for it to do so (section 5.2.4.2), based on the information currently held in its mental model. This eliminates the problems just mentioned, as long as the mental model is sufficiently complete (section 4.3.5.2); if this is the case, an overtaking manoeuvre will never have to be abandoned as it will have been planned in advance.

During the actual overtaking manoeuvre an actor's speed is boosted, to limit the amount of time that it is out of its normal lane of travel; this is an observed real-world effect.

If an actor is unable to overtake, it will be forced to slow down until the opportunity arises; in this way lane formation once again emerges.

6.3.2 Improvements to eye gaze behaviour

One of the most important novel areas in this simulation is a psychologically accurate simulation of vision and attention. The effects of this simulation upon the behaviour of actors will be discussed in section 6.3.3, but it also leads to a number of improvements in the eye gaze behaviour of each actor.

In many previous semi-realistic 3D simulations of virtual actors no consideration has been given to the eye movements of the actors. In some cases the models used have been insufficiently detailed to enable the direction in which the eyes are looking to be perceived; alternatively, actors may never be seen in close-up: often, only an overhead view will be seen, in which the eyes are naturally not visible. In systems where actors' eyes are visible, they will either be fixed to look straight ahead, or may move around randomly.

As the quality of 3D models and rendering techniques improve, crowd simulations can increasingly be used for the foreground of a scene in a film, rather than just filling in background. (For example, in *The Fellowship of the Ring* (2001) crowd simulations were only used for horses and riders at a great distance from the camera, with real actors in the foreground. By *The Return of the King* (2003) actors filling half the screen were part of a crowd simulation.) Therefore it is necessary to produce an accurate simulation of eye movements, as the gaze direction will be clearly observable in this type of close-up. Research by Colburn, Cohen and Drucker [CCD00] has shown that accurate eye movements are a crucial part of the process of deciding whether an actor looks realistic or not.

The eye movements of the actors in this system are produced as a direct result of the attention simulation system (section 4.2.1), rather than requiring separate, specialised rules. As well as being simpler, this also produces more accurate results: the purpose of attention simulation is to determine the current target that the actor is concerned with, and this should be reflected in its gaze direction.

A number of insights into an actor's behaviour are made visible in their eye gaze direction:

- If the actor is particularly concerned with another actor, it will fix its gaze on it, tracking it as it moves (using simulated pursuit movements).
- If a number of actors are of concern, it will alternate between them, flicking its gaze direction to and fro.
- If no actors are of particular concern, it will be seen simply looking around, seemingly at random, continually scanning its surroundings.
- The actor's emotion may be determined from its gaze, depending on whether it is looking all around, looking mainly straight ahead, or checking behind – see section 7.3.1.

In other words, the simulation of eye gaze behaviour allows a viewer to empathise more with the actor, as well as adding to the realism. Examples of these types of eye gaze behaviour were given in section 6.2.1.2.

6.3.3 Effects of attention simulation on behaviour

Although attention simulation leads to the improved eye movements just discussed, the principal reason for performing the simulation is to study its effects on the behaviour of the actor, in the hope of producing a more accurate and believable re-creation of human behaviour. This section examines the alterations in behaviour resulting from the use of attention simulation, and determines whether they lead to a more realistic result.

Effects of attention simulation can be divided into those which affect the behaviour of an individual actor, and those which produce overall changes in the behaviour of the crowd. Further discussion of the mental model was also presented in section 4.3.5.

6.3.3.1 Individual behaviour

Collisions The designed limitations in each actor's mental model, discussed in section 4.3.5.2, lead to an increase in the number of collisions, such as in the example shown in section 6.2.1.2 [Ignoring others]; these are caused by an actor either being unaware of the presence of another actor, or having an incorrect idea of its position or velocity. In the most extreme cases, if both actors are completely unaware of each other, there may be an actual physical collision. However, it is more normal for the presence of the colliding actor to be noticed right at the last moment, when it is too late for any avoiding action to be taken; in these cases both actors will stop before actually coming into contact. Conceptually, this can be treated in the same way as an actual collision, as this also involves both actors coming to a complete stop.

Section 6.2.2.1 [Number of collisions] presented quantitative results showing the effect of attention simulation on the number of collisions occurring: using attention simulation produces roughly a 60% increase in the number of collisions, compared with giving all actors total awareness of those around them. A cause for concern was whether this increase was simply due to actors now having *no* awareness of those around them; however, it was shown that this would in fact lead to around a 135% increase in collisions.

Therefore it is clear that the number of collisions resulting from attention simulation is definitely between the two extremes of total and zero awareness – actors have *limited* awareness of their surroundings.

It might appear that increasing the number of collisions is not an advantage. However, it does in fact produce a simulation that is a better approximation to reality, for the following reasons:

- If a person is looking over his shoulder, or distracted in some other way, while walking forwards, he may well collide with another person (or object): without a simulation of awareness, this could not happen. An example of this was shown in section 6.2.1.2 [Distractions].

- If actors have total awareness then collisions will be predicted at an early stage and a detour could be started in plenty of time to cleanly avoid the collision. This does not reflect the real world, where collisions will often be predicted late on, and will then be avoided either by sudden movements, or by the step-and-slide. In fact, with total awareness there would never be any need for actors to use the step-and-slide (other than based on their mood), when it is in fact a very common feature of pedestrian behaviour.

- In the real world, a number of collisions do occur, as collision avoidance is not perfect: the majority of these are minor, and perhaps only involve a very small amount of contact. In addition, there are a large number of near misses, avoided by one person swerving at the last moment. With total awareness, these would never be re-created – with attention simulation they are.

- If actors are given total awareness, then they will know exactly how far they need to detour to avoid a collision taking place. This is unrealistic: people cannot predict exactly where somebody else will be at the point of collision, and will therefore allow a safety margin – they will detour more than necessary. With the use of attention simulation, actors will often have inaccurate information about the position and velocity of others, and will make detours that are larger than required.

In summary, giving actors total awareness leads to perfect collision avoidance behaviour: collisions will nearly always be avoided, avoidance behaviour will start in plenty of time, and actors will only alter their courses by the smallest possible amount. Although this might seem technically pleasing, it will look artificial – the actors will look more like robots than people. Attention simulation results in more chaotic behaviour that is a better approximation to reality.

Overtaking Inaccuracies in an actor's mental model may lead to problems when it is overtaking. As described in section 6.3.1.3, one of the novel features in this system is the check an actor will make to see it is safe to overtake. However, an actor is only able to access data from its mental model to perform this check. It is for this reason that, before making the decision, an actor will make an explicit request to look in the direction it is planning to travel, to check for the presence of any undetected actors. Despite this check, an actor may appear after overtaking has commenced, blocking the planned path. There are several ways an actor may not be detected in advance:

- It may have been too far away to be seen, but moving fast enough to become a problem.

- It could have been occluded, either by other actors or by an obstacle.

- It could be present in the mental model, but with inaccurate, out of date data, thereby not appearing to present a problem.

In other models of vision, such as total awareness, all obstructing actors would be detected in advance. Therefore the use of attention simulation will lead to more cases where overtaking is commenced, and then has to be abandoned due to the sudden detection of an obstruction.

Conversely, inaccurate information in the mental model may cause overtaking not to occur, even though it would in fact be possible, if an actor is incorrectly thought to be an obstruction.

This is another example of attention simulation causing behaviour to be less perfect, but more realistic.

6.3.3.2 Crowd behaviour

The effects upon behaviour just described will also combine to produce changes in the overall behaviour in the crowd. They will affect the overall speed of movement of the crowd, which will have an effect upon the time taken to evacuate a building; more formation of lanes may also result.

Speed of movement When an actual collision occurs, involving some sort of physical contact between actors, both of them will immediately come to a stop, before negotiating their way around each other. Even if there has not been any contact it will often still be necessary for the actors to stop walking to avoid bumping into each other. Alternatively, if a potential collision has only been detected late on, then actors may be forced to slow down in order to avoid the collision.

Therefore, if the number of actual collisions and near misses increases then actors will be stopping or slowing down more frequently. This in turn has an effect upon the overall speed of movement of the crowd, in two ways: there is a direct effect, as if an actor is often stopping it will have a slower average speed compared to walking continually at its normal pace. In addition, there is an indirect effect: if one actor stops or slows down, any actors walking behind it may also be forced to stop, to avoid a collision. In this way one collision taking place can cause a ripple of other actors being forced to slow down.

As has already been discussed (section 6.3.3.1), experimental evidence obtained from this system (section 6.2.2.1 [Number of collisions]) showed that the number of collisions increases by around 60% when using attention simulation, compared with giving actors total awareness. This increase will therefore tend to cause the crowd as a whole to move significantly slower.

A particular example of this can be seen when a large number of actors are all trying to pass through a narrow doorway or gap. Collisions will occur, particularly due to each actor's difficult task of trying to monitor the positions of large numbers of others simultaneously. As these collisions occur the actors involved will stop, and may block the entire doorway; until they resolve their collision, no other actors will be able to pass through.

Evacuation times The accurate simulation of the speed of movement of a crowd is especially important when evacuations are being simulated to calculate how long it would take people to leave a building; for example, there may be a requirement that everybody should be able to evacuate in under five minutes.

The results presented in section 6.2.2.1 [Time to evacuate] show that, as expected, evacuation time increases by a significant amount – 54% – when using attention simulation as compared to total awareness. For example, an evacuation that was predicted to take five minutes, might in fact take almost eight minutes.

The reasons for this increase in time are those just mentioned: individual actors will move more slowly due to the effect of collisions, and this will have an impact upon other actors surrounding them. Collisions blocking doorways will also have a contributory effect.

It is clear therefore that to get accurate results it is essential to consider the effects of attention, as simulated in this system.

Lane formation One of the effects of the mental model system is that it requires actors to make explicit requests to look in the direction they are planning to use when overtaking; these must be completed before the manoeuvre can be started. If the distance between an actor and its collidee in front of it is small (as will often be the case when a collision is predicted, due to incomplete information) then the actor will have to slow down to the speed of the collidee while this request is fulfilled, to avoid a collision with the collidee. This is the same behaviour as an actor will perform if they have chosen to slow down, rather than overtake. Therefore, this system will lead to more lane formation, some of it on a temporary basis.

This is in addition to the increase in lane formation due to actors being given the option of slowing down rather than overtaking, discussed in section 6.3.1.3 [Lane formation].

6.3.4 Uses of the system

The experiments carried out in section 6.2.2.3 on different layouts of doors demonstrated one of the ways in which the more accurate behaviour simulated in this system can be used to test real-world scenarios. In that example it was demonstrated that having two narrow doors side by side was more efficient at evacuating a crowd of people than having one wide door. Careful study of the simulation shows the reason for this: the single door has to be used by actors coming from both right and left sides of it. Due to inaccurate mental model data, there will be a number of head-on collisions or near-misses between these two streams of actors. This will lead to a slowing down of actors within the doorway, causing it to be temporarily blocked on a number of occasions – during which time no actor would be able to exit through it. However, when there are two doors, actors coming from the left will nearly all make use of the left-hand door, leaving the right-hand door for those from the right. Because most of the actors using each door will be moving in the same direction, the number of collisions will be dramatically reduced: both doors will have free-flowing streams of people, allowing evacuation to proceed quickly.

Similar experiments could be performed by a user of the system, involving many different types of door, corridor and building layout, up to simulations of entire buildings. The use of attention simulation means that a higher confidence may be invested in results obtained from these simulations.

6.3.5 Summary

The preceding sections have demonstrated that the simulation of attention used in this system has a significant effect upon the behaviour of actors. It leads to an increase in the number of collisions and near-misses that an actor will experience, and results in overtaking not always being successful. On a large scale it causes slower crowd movement, leading to longer evacuation times, and to more lane formation.

These may not all seem like desirable effects – are more collisions and slower movement actually a good thing? However, it is this author's belief that it is features like these that make the simulation a better re-creation of real human crowd behaviour, and less like an artificial computer simulation. In comparison with the smooth, carefully planned movement often seen in simulations, real behaviour consists of people often being forced to slow down or stop, getting stuck in queues or jams, changing course suddenly, using the step-and-slide, and having minor collisions with other people. All of these features are re-created in this system, contributing to an accurate, believable simulation.

Chapter 7

Moods and Emotions

The previous chapters have described the attention and behaviour systems that are applied to each actor in the simulation. However, without a simulation of moods and emotions, all the actors will behave in exactly the same way when faced with a particular situation. The work described in this chapter aims to rectify this situation.

This chapter starts with the selection of emotions to simulate, followed by a description of the ways in which each actor's mood is defined, then discusses the specification of the relations between mood and behaviour. Each behaviour variation will then be covered in detail, before a few implementation details are discussed, and examples presented.

7.1 Emotion selection

It was decided that eleven of the emotion categories defined by Baron-Cohen et al. [BCGWH04] (section 3.3.1) could potentially be simulated in this system, as they have an observable effect upon walking behaviour or gaze movements. Ten of these emotions were found to fairly readily form into pairs of opposing emotions:

- Thinking – Excited
- Bored – Interested
- Sad – Happy
- Unfriendly – Kind
- Unsure – Sure

This leaves Afraid without an opposite, so one of the sub-emotions from Happy was selected for this purpose – Carefree. Although this was not listed as an emotion category, it seems sufficiently different from Happy to be worth simulating separately.

Finally, two extra moods were added: Strolling and Hurrying; they do not have an effect upon facial expression, which would explain their omission from Baron-Cohen's set of emotions. However, they will clearly have a large effect upon walking behaviour, which would seem to justify their inclusion.

7.2 Initialisation

7.2.1 Actor mood setup

Before the simulation can be started each actor is initialised with a set of moods and emotions, which can later be altered during the progress of the simulation.

Every actor stores an integer for each of the mood ranges defined in section 7.1, between -100 for one end of the range up to 100 for the other end. For example, an actor who was fairly afraid and significantly hurrying would have a value of -30 for the Afraid–Carefree range, and 80 for the Strolling–Hurrying range. This value range was chosen as it means integers can be used, allowing efficient processing, while still giving a suitably large number of possible settings.

Actors are initialised with either a random set of emotions, or the default value (zero) for each range. A graphical user interface is provided for the user to alter these settings (section 7.4).

7.2.2 Mood-behaviour table

The links between moods and behaviour need to be codified so that, for any mood values, the desired effect upon behaviour can be determined. For example, if an actor is slightly Afraid and very Unsure, what effect should that have upon its walking speed? To solve this problem, the system contains a mood-behaviour table.

7.2.2.1 Design of the mood-behaviour table

The mood-behaviour table has the 14 possible moods on one axis, and the 12 behaviour variations (to be described in section 7.3) on the other. An entry in the table represents the amount and type of effect a mood should have upon

	Afraid	Carefr	Thinki	Excite	Bored	Intere	Sad	Happ	Strolli	Hurryi	Unfrie	Kind	Unsur	Sure
Looking around	100	-50	-100	100	-100	100	-25	0	50	0	-100	25	100	-50
Looking far away	100	-75	-25	0	0	100	-100	0	25	-25	0	0	50	-25
Monitoring often	100	-75	-25	0	-50	100	0	0	0	50	0	0	50	-50
Looking behind	100	-100	0	0	0	25	0	0	0	0	0	0	50	0
Large detours	100	-50	-25	0	-50	0	0	0	25	-100	-100	75	50	-50
Walking quickly	75	-50	-50	50	0	0	0	0	-100	100	0	0	-25	25
Overtaking preferred	0	-50	-25	25	-25	0	-25	25	-75	100	50	-25	-50	50
Eye contact	-100	50	-75	25	-50	100	-100	100	25	-50	-100	50	0	0
Stopping to talk	-100	25	-75	0	-75	100	-75	100	50	-100	-100	25	0	0
Take risks	50	0	0	0	0	0	0	0	-100	100	0	0	-75	75
Avoid early	75	-25	-75	0	-25	0	-25	25	50	-75	-100	100	75	-75
Do not avoid	-100	0	0	0	0	0	0	0	-100	25	100	0	0	0

Figure 7.1: The mood-behaviour table

a behaviour variation. Integers between -100 and 100 are used, for the same reason as above. The table is shown in figure 7.1.

In the table:

- A value of 100 means the mood will have a significant effect upon the behaviour variation. For example, whether or not people are afraid will have a significant effect on whether they look behind themselves; the more afraid they are, the more likely they are to do so.

- A value of -100 means the mood will have a significant effect on the opposite of the behaviour variation. For example, the more unfriendly somebody is the *less* likely he is to make large detours around others.

- A value of 25 means the mood only has a small effect upon the behaviour variation. For example, the kinder people are, the more they will look around themselves, but only to a limited extent (whereas for the afraid emotion the value for this behaviour would be 100, as it has a significant effect).

The values in the table for the opposing halves of the seven pairs of moods are clearly linked: they should generally be very different – the behaviour for somebody who is sad will be the opposite from somebody who is happy. Specifically, the two values should be one positive and one negative, one zero, or both zero (in other words, not both positive or both negative). These requirements are set so that the range passes through zero; this allows a zero mood value to be mapped to a zero effect upon behaviour, as described later. This means that if all mood values are set to zero then the actor will behave in a default manner, with no emotions present.

7.2.2.2 Values in the mood-behaviour table

The default settings for the mood-behaviour table values are based on the results of the interviews described in section 3.3.2 (but can be altered by a user of the system). The process of converting interview responses into numeric form required some thought as participants clearly did not mention every single connection between moods and behaviours that is needed to create the table.

- Where links between emotions and effects were mentioned frequently, they were entered into the table with a large value; for example, nine out of ten participants mentioned that people who are Afraid will look around a lot, so that link was given a value of 100.
- Pairs of emotions were always given values indicating opposite effects; for example, looking around while Carefree was given a value of -50, even though this was not often explicitly mentioned.
- Some values were created by 'interpolation' from others. For example, nobody mentioned that an Excited actor might make eye contact with others, but this was mentioned frequently for a Happy actor; given the similarities between these emotions, a small value (25) was given to Excited.
- This left a few values which were filled in mainly through common sense and personal opinions of the author.

7.2.3 Actor behaviour setup

Once each actor has a set of mood values, and the links between mood and behaviour are defined, the values for each actor's behaviour variation settings can be calculated. These behaviour-variation values control how much that variation will be applied; for example, if the behaviour variation 'Large detours' is set to -100, the actor will make very small detours, if to 100, it will make very large detours.

The general idea is that the behaviour-variation value should have a maximum of 100 and a minimum of -100, and the effect of a mood should be proportional to its value in the mood-behaviour table for that behaviour variation. Full details of the technique used are given in appendix A.5.

7.3 Behaviour variation

The moods and emotions of an actor have an effect on many different aspects of its behaviour, both in terms of eye movement and also its walking behaviour. The source for most of these behaviour variations was the interviews described in section 3.3.2, as well as the psychology literature discussed in section 3.3.1.

The 12 behaviour variations are now discussed in detail, along with explanations of how they have been implemented in this system.

7.3.1 Eye movement variation

There are five aspects of eye movement that can be altered based on mood:

- Looking around: affects how much an actor will look all around itself, looking in many different directions rather than just in its direction of travel.
- Looking far away: determines how far away actors will be detected, representing the fact that some people only look at those close to them; this is particularly true for people that are looking down at the ground.
- Monitoring often: specifies how often the actor should perform repeated observations on actors it has already seen.
- Looking behind: sets how often, if at all, the actor should check over its shoulder to detect actors behind it.
- Eye contact: controls how often actors will hold eye contact with others around them.

7.3.1.1 Looking around

The variation caused by this behaviour is introduced when an actor's attention manager is deciding which is the most urgent actor to look at next within each of the categories of looking: explicit requests, periphery capture, monitoring and spontaneous looking.

Explicit requests and periphery capture The effect of this behaviour is to restrict the directions in which an actor will look, by diminishing its potential field of view. If the value is zero or positive then there is no restriction compared to normal behaviour – the actor may look at anything in front of it. However, if the value is negative then the field of view is restricted to a range between 5°

(for −100) and 90° (for 0) either side of the actor's current direction of travel. Actors will ignore any explicit or peripheral vision requests outside this range, preferring to look mainly straight ahead.

Monitoring If the behaviour-variation value is positive, the potential field of view within which actors will be monitored ranges from 60° (for 0) to 90° (for 100) either side of the direction of travel; this extends the field of view beyond its normal values if the actor is particularly inclined to look around. For negative values the field of view is restricted to between 60° (for 0) and 5° (for −100).

Spontaneous looking Rather than just restricting where the actor can look, if it has a positive behaviour-variation value it should actively look all around the scene, by performing large gaze shifts while conducting spontaneous looking. This is implemented by setting a preferred angle shift – a fraction of the maximum possible angle shift (while still looking forwards) in proportion to the behaviour-variation value. The target requiring a gaze shift closest to this angle is initially selected. If the behaviour-variation value is zero, the result is as normal – the actor requiring the smallest gaze shift from the current direction is selected.

For a positive behaviour-variation value, no further action is needed. However, if the value is negative, the actor's choice of targets is restricted, as it should not be looking around too much. A maximum allowed angle change is calculated, a proportional percentage of the maximum possible angle change; if the angle change is greater than this, spontaneous looking will be abandoned.

Summary Negative values will restrict the directions in which an actor will look, up to extreme situations of only looking 5° either side of its velocity. Positive values cause no restrictions, and encourage the actor to look all around the scene when it can.

7.3.1.2 Looking far away

The effect of this behaviour is simply to restrict how far away an actor would be noticed; the restriction is applied to all categories of looking in exactly the same way. The maximum allowed distance is a value between 2 m and 30 m (the maximum given by Wolff (section 3.1.3), linearly proportional to the behaviour-variation value. Only actors nearer than this distance will be considered; if no actors are suitable then the subject will simply look straight ahead.

In addition to simulating the effect of people only bothering to look at those near them, this also approximates the idea of people looking *down* – this idea was frequently mentioned in the interviews (section 3.3.2). The principal effect of looking down is that only nearby people will be visible, at the top of the person's vision. Therefore, an actor who is sad will have a negative value for this behaviour. (This also makes the arrow representing its gaze point downwards.)

7.3.1.3 Monitoring often

This behaviour variation affects the frequency with which an actor will look at those actors it has already seen.

Under normal circumstances, with no emotions, the decision about which actor to monitor is based on two factors: time to collision and how overdue the actor is for re-observation (see section 4.3.1.3). In particular, if no collision is predicted for a particular actor, and it is not overdue for re-observation, it will be ignored; if all potential targets for re-observation fit this pattern, spontaneous looking will be performed rather than monitoring.

This procedure is altered if the behaviour-variation value is greater than zero: when an actor is found that would normally be ignored there is a chance, linearly proportional to the size of the value, that it will in fact be considered for observation. (The test checks if a random number between 0 and 100 is less than the behaviour-variation value – all random numbers in the system are uniformly distributed.) Priority is still given to actors that have not been observed for the longest period of time.

This has the result of actors being more aware of the positions of those around them, if they have high values for this behaviour. If they are monitoring multiple actors they will cycle between looking at each of them; if only one actor is being monitored, it will be looked at very frequently – constantly if the behaviour-variation value is 100.

Negative values of the behaviour-variation value have no effect, although negative values from some emotions can counteract the effect of positive values from others.

7.3.1.4 Looking behind

The 'Looking behind' behaviour is principally used when an actor is 'Afraid', when it wants to look over its shoulder to check that there is nobody following it who could be a threat. This behaviour affects periphery capture, spontaneous looking and monitoring.

Periphery capture and spontaneous looking Normally the maximum field of view from which targets can be selected to be observed is set at 90° either side of an actor's velocity – the area in front of the actor (although this may be reduced based on the value of the 'Looking around' behaviour). To allow the actor to detect targets behind it, this field of view needs to be increased. A test, linearly proportional to the behaviour-variation value, decides whether the field of view should be extended to 180° either side of the actor's velocity, thereby allowing it to look at any target behind it. In other words, the greater the behaviour-variation value, the higher the chance that a target behind the actor may be observed.

Monitoring The situation for monitoring requests is a little more complicated, as using the above system could cause an actor to almost continually look behind itself, which would look unrealistic: people do not walk along continually looking behind themselves, rather they make a number of quick glances. This effect is achieved in several ways.

An actor will only have the option of looking behind on one in every three decisions about where to look. Simple trials were carried out, altering how often this decision could be made, and observing how many looks behind occurred. If the actor was allowed to make the decision more frequently then high behaviour-variation values would cause them to look behind almost continually, which is unnatural; if the decisions were less frequent, actors would not look behind enough even with high behaviour-variation values. The setting of one in three was found to balance the desire to look behind with still making a number of looks forward.

When an actor has the option of looking behind, a test is performed, proportional to the *square* of the behaviour-variation value. (Negative values are ignored. Otherwise, the value is divided by ten, and then squared, meaning it stays in the range 0 to 100. This value is then compared to a random number between 0 and 100.) Once again, this technique was the result of observing how often an actor would look behind; a simple linear proportionality was found to give too many looks behind for a mid-range value, whereas this technique only produces a large number of looks behind for very high behaviour-variation values.

Summary Negative values have no effect (although negative values from some emotions may cancel positive values from others). Small positive values will cause an actor to occasionally detect new actors behind it, and very occasionally continue to monitor them; large positive values mean new actors are observed wherever they are, including behind, and all actors will be monitored, although there will still be fewer backward glances than forward glances.

7.3.1.5 Eye contact

People who are 'Interested', 'Happy' or 'Kind' are likely to make and hold eye contact with others around them, in addition to the normal monitoring that takes place purely for the purpose of working out position and velocity. On the other hand some people, particularly those who are 'Afraid', will do everything they can to avoid eye contact being made, quickly looking away if it occurs.

This behaviour is implemented by adding eye contact as an option to perform instead of spontaneous looking if the behaviour-variation value is positive, and imposing a restriction on which actors may be looked at if the value is negative.

Eye contact option In situations where an actor would normally perform spontaneous looking, once it had exhausted all other options, there is a chance, proportional to the behaviour-variation value, that it will instead continue to look at the actor which it observed last. This will only occur if the actors are walking in opposite directions and towards each other, so that they can both see each other. The actor will continue to look at the same actor until either the chance fails, or a more urgent call on attention is found.

Eye contact restriction If the actor's attention model has selected the same target for the actor to look at as on the last frame there is a chance, proportional to the behaviour-variation value, that it will be forced to look away instead: it will be given a default value of looking straight ahead.

Summary There are very different behaviour patterns at opposite ends of this scale: some actors will continue to look at others for a long time, while others will only ever cast quick glances at others. While this is only a simple re-creation of real-world behaviour, it is a worthwhile approximation given the importance of this behaviour. Further ideas are discussed in section 8.2.6.

7.3.2 Walking behaviour variation

There are seven aspects of actors' standard walking behaviour, as described in chapter 5, that can be varied based on mood:

- Walking quickly: sets the actor's default speed.

- Large detours: determines how much space the actor will leave when avoiding a collision with somebody else.
- Avoid early: specifies how early the actor should start avoiding a collision.
- Overtaking preferred: sets how little the actor is willing to slow down to avoid having to overtake a collidee.
- Do not avoid: assigns a probability that the actor will ignore a predicted collision and just keep walking.
- Take risks: determines whether the actor will engage in 'risky' behaviour to get to its goal faster.
- Stop to talk: sets the probability that an actor will stop to talk to other similarly inclined people.

7.3.2.1 Walking quickly

The speed at which an actor walks is primarily determined by whether it is 'Hurrying' or 'Strolling', but other emotions also have an effect: actors that are 'Afraid' or 'Excited' will also walk quickly.

Each actor has a preferred speed at which it will walk unless it has a reason not to, such as overtaking or slowing down to avoid a collision. Rather than setting this randomly, as in the emotion-free system, it is set to contain a part proportional to the behaviour-variation value, and a random part. The behavioural part is linearly proportional to the behaviour-variation value setting, and ranges from zero to 1.3 m/s; the random part varies between 0.6 and 1.2 m/s. This gives a total possible range of 0.6 to 2.5 m/s. This is 0.3 m/s wider at each end than the range observed by Daamen and Hoogendoorn [DH03] (section 3.1.2) of 0.9 to 2.2 m/s. This was a deliberate choice, as it seems certain that, in extreme situations, such speeds must in fact be possible; however, because they are rare, it is quite possible that they would not be seen during an experiment.

The reason for the two components of the preferred speed is to ensure that, while it does depend mainly on the behaviour-variation value, not all actors with the same value will have the same speed – there is a total possible variation of 0.6 m/s for actors with the same value. The behaviour-variation contribution is approximately twice as significant as the random contribution.

7.3.2.2 Large detours

Actors who are 'Afraid' are likely to give others a wide berth when avoiding them, as they do not want to get too close to them. Actors who are 'Kind' will also perform large detours around others, but this is instead out of politeness. Conversely, actors who are 'Hurrying' or 'Unfriendly' will make the smallest detour possible.

This behaviour variation is implemented in two places: collision prediction and collision avoidance.

Collision prediction During normal collision prediction, described in section 5.1, a default value is used for the radius of the personal space around each actor; this is set to 0.3 m. If this value is altered then it affects how much space an actor leaves when avoiding a collision; for example, if it is increased then a route that previously would have been acceptable will now be predicted as being a collision.

To this end, if the behaviour-variation value is negative – the actor wants to perform small detours – then the personal space radius is decreased, down to a minimum of 0.2 m; values any lower than this would cause actual collisions not to be predicted. If the value is positive then the personal space radius is linearly increased up to a maximum of 1 m for the highest value; this forces the actor to make very significant detours.

Collision avoidance An actor wishing to make large detours will need to make larger movements to get further away from the path of its collidee. This is implemented by making the actor prefer larger angular direction changes, while still leaving open the possibility of smaller changes if other paths are blocked.

In particular the 'Direction change' and 'Direction and speed change' functions described in sections 5.2.3.1 and 5.2.3.3 are altered so that they do not start by trying a 10° angle change. Instead the starting angle is on a range between 10° and 60°, in proportion to the behaviour-variation value. Starting with this value increasingly large changes are tried, as before, but there is also the possibility of jumping back to try smaller changes if required.

7.3.2.3 Avoid early

Actors that are making large detours around others will often combine this with another behaviour: starting to avoid the collision earlier than normal. This makes the process easier, as there is more time for the actor to get further out of the way. In normal circumstances these behaviour-variation values will be similar, but they are kept separate as there are some combinations of emotions that would give them different values. For example, an actor who is thinking hard will avoid very late (due to the −75 entry in the mood-behaviour table), but there is little effect upon the size of the detour (−25). By contrast, a bored actor will make deliberately small detours (−50), but there is less effect on when avoidance starts (−25).

As soon as a collision has been predicted the actor will start trying to avoid it. Therefore, one way of varying the time the actor starts its avoidance behaviour is to vary the time at which a collision prediction is reported.

In the system described in section 5.1 a collision was reported if the distance between actors was less than 15 m. This is changed to a range between 2 m (for the most negative behaviour-variation value) and 30 m (for the most positive value). This gives a very wide range, from only finding out about the collision at almost the last moment and having very little time to prepare, to a collision being avoided well in advance, at the maximum end of the range given by Wolff (section 3.1.3).

This alteration also necessitates a change in the routine for detecting if a Towards collision is over (section 5.3.1), in which one way of classifying a collision as over is if the actors are further than 4 m apart. If the actor is avoiding particularly early, this may now be before the avoidance has actually happened. Therefore this value can be increased to up to 30 m, in proportion to the behaviour-variation value.

Summary The two behaviour variations of 'Large detours' and 'Avoid early' combine to give a range of behaviours varying from very timid actors that get out of the path of a collision early on, and go significantly out of their way to put a large distance between themselves and the collidee, to confident actors that only move out of the collision path at the last moment, and only move just enough to avoid an actual collision occurring.

7.3.2.4 Overtaking preferred

The simple, interim system used to determine whether actors would slow down or try to overtake to avoid an Away collision was described in section 5.2.4. In the Moods and Emotions framework this is replaced with a separate behaviour-variation value: 'Overtaking preferred'. Actors that are 'Hurrying' will clearly prefer to overtake, as will actors who are 'Sure'.

The first step in deciding whether to overtake or not is to work out how much the actor would have to slow down to walk at the same speed as its collidee; this is calculated as a percentage of its current speed. The behaviour-variation value is scaled to a value between 0% (for 100) and 90% (for −100): this is the amount the actor is prepared to slow down; if the amount it would need to slow down is more than this, it will choose to overtake. A slowdown of any more than 90% is considered unreasonable and not something that any person would be prepared to do; this is an ad hoc value, but appears to match real behaviour.

Summary This behaviour can be used to re-create the real-world effect that most people will not mind slowing down a little rather than having to go to the effort of overtaking, but will not be willing to make large changes to their speed. Actors that are 'Hurrying' will be prepared to make only very small changes to their speed, nearly always preferring to overtake, while at the other extreme some actors will make every effort to avoid overtaking.

7.3.2.5 Do not avoid

If a person is 'Unfriendly' or, to a lesser extent, 'Hurrying', then he will not make any effort to avoid Towards or Glancing collisions, preferring instead to 'barge through' – pushing other people out of the way. He will either hope that his collidees will perform sufficient avoidance behaviour on their own and get out of his way, or he will perform the step-and-slide manoeuvre (section 5.2.3.4) at the last moment.

Each time the actor would normally perform collision avoidance there is a chance, proportional to the behaviour-variation value, that it will not do so. If the value is set to 100 the actor will never perform collision avoidance; if it is set to 50 then it will only perform avoidance on half of the *frames* – not on half of the collisions. This means that an actor with an intermediate value will begin to move out of the way, but not by a sufficient amount to avoid the collision by itself. This has the desired effect of simulating some sort of laziness on the part of the actor, rather than occasional decisions to completely ignore others (which would be the effect of preserving the choice for the entire progress of the collision).

An actor will always try to avoid an Away collision, whatever the setting of this value, as it is generally unlikely that the actor will be able to barge past the collidee from behind. Instead, other behaviour-variation settings will usually mean that the actor will try to overtake, making the shortest detour possible.

7.3.2.6 Take risks

A person who is ‘Hurrying’ may decide to attempt to get to his destination more quickly by engaging in risky behaviour; somebody who is ‘Afraid’ may do the same, to get out of the way of somebody following him, as may somebody who is very ‘Sure’ of what he is doing.

For the purposes of this simulation, the example of risky behaviour chosen is the decision about whether to take a short-cut across a road, rather than staying on the pavement or using a designated road-crossing area.

In normal operation, an actor is stopped from walking on the road in the same way it is for any obstacle: firstly by checking any collision avoidance behaviour to ensure that it will not require walking on a road in the near future, and also by actively turning back actors that attempt to walk on a road. Both of these procedures need to be altered.

Each actor is given a new field known as a ‘road pass’: if an actor possesses this pass, it is allowed to walk on the road. The pass is allocated when the decision is first made to let it walk on the road, and is revoked when it finishes walking on the road. This scheme means that once an actor has started walking on a road it will continue to do so, rather than suddenly turning back. (Avoidance of road traffic is not considered in this system.)

Whenever an actor checks to see if a collision avoidance behaviour would cause it to collide with an obstacle, there is a chance, proportional to the behaviour-variation value, that it will be allowed to ignore roads in this check. If, on these occasions, the avoidance behaviour is acceptable, apart from the presence of a road, it will be given a road pass, and allowed to use that route.

When a collision between an actor and an obstacle is predicted, a check is made to see if that obstacle is a road. If it is, and the actor has a road pass, the collision will be ignored. Alternatively, if it does not have a road pass, there is a chance, proportional to the behaviour-variation value, that it will be given one, and will therefore continue walking.

Summary Actors with high behaviour-variation values will often make use of the road when avoiding Towards collisions, and when overtaking others; this allows them to continue walking quickly whereas otherwise they would have to slow down to avoid the collision.

7.3.2.7 Stop to talk

People who are ‘Happy’ or just ‘Strolling’ along will regularly stop and talk to people that they meet – mainly people that they already know. This is such a common feature of the real world that any attempt to simulate it will improve the realism of the system.

On each frame of the simulation there is a 20% chance that the system will look for actors that might want to stop and talk. This figure was chosen using trial and error, by observing how often the behaviour occurred with different chances; the aim was to ensure that there is not an over-abundance of stopping, even when actors have high behaviour-variation values.

If this condition is met then each actor is examined in turn, and its behaviour-variation value is compared to a random number to give a higher probability of stopping if an actor has a high value. If this actor does want to stop, all the actors it is currently monitoring are studied; if any one is:

- less than 1 m away,
- walking in the opposite direction from the actor,
- facing the actor, and
- wanting to stop (its behaviour-variation value is also greater than the random number),

then that actor is selected as the one to stop and talk to. Both actors are stopped, and turned to face each other. They will stop for a random length of time between 20 and 400 frames – one second (a quick "Hello") to 20 seconds (a brief chat). After this time has elapsed, both actors will continue on their previous course.

Summary Actors with high behaviour-variation values will often stop to talk to those actors with similarly high values, while other actors will mostly ignore each other; any actors with negative values will never stop – this would include actors that are 'Afraid'.

This behaviour is clearly a great simplification of real human behaviour, as people normally only stop to talk to people they know, rather than just other talkative people; however it does at least give the impression of sociable people.

7.4 Implementation

This section briefly describes how emotion controls have been added to the user interface described in section 6.1.

Nearly all of the options available to control the moods and emotions of the actors are located on a separate pop-up window which, ideally, can be placed

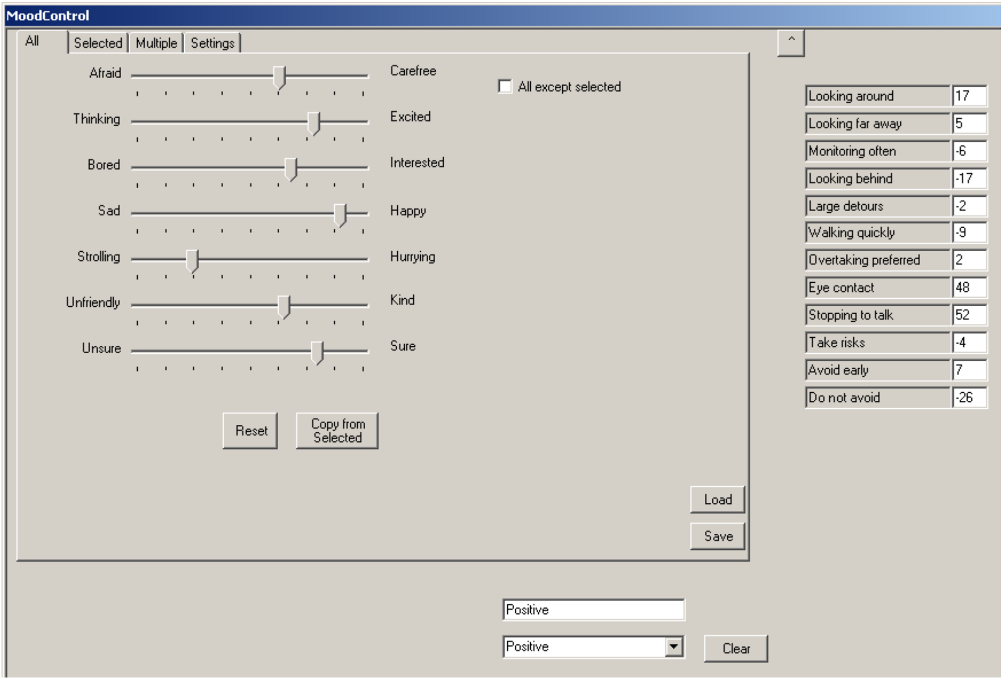

Figure 7.2: The moods and emotions user interface

on a secondary monitor; this window is shown in figure 7.2. It contains four tabs:

1. A tab containing seven sliders, for each of the seven ranges of emotion (e.g. 'Sad' to 'Happy'); each slider ranges from -100 to 100. These emotions will be applied to *all* the actors in the scene uniformly, although selected actors can be exempted. This tab is useful for setting up the default behaviour of actors, before specific changes are made to others. Emotion settings can in addition be loaded and saved. As the sliders are moved the behaviour-variation values displayed to the side are automatically updated.

2. A tab almost exactly the same as tab 1, used to set the emotions of the currently selected actors. The values from tab 1 can be copied here, so that small changes can be made to the default behaviour. In addition, a slider allows a number of actors to be selected at random; this means a user can set, say, 25% of the actors to be 'Hurrying', without caring which 25% that is.

3. A tab that allows the user to load and save complicated emotion setups. For example, the user could create a setting whereby all the actors are

'Friendly', 25% of them are 'Hurrying' and 75% of them are 'Strolling'; this can then be saved for future use.

4. A final tab which allows the user to edit the values in the mood-behaviour table (section 7.2.2).

Two extra drop-down boxes were also added to the main simulation window. The first contains all the saved emotion settings from tab 1, the second the saved settings from tab 3; both allow presets to be applied quickly.

7.5 Results

This section presents results of the evaluation of the moods and emotions component of the system. First, pictures and videos of the system in action are presented; then the results of user testing conducted during a video demonstration are given.

7.5.1 Graphical testing

Each of the 14 emotions simulated in this system is part of a pair; the seven pairs were listed in section 7.1, each pair representing two extremes of emotion. Therefore, a useful way of demonstrating the differences in behaviour produced by an emotion pair is to assign one of the extremes to a proportion of actors, and the other extreme to some other proportion. In all of the pictures and videos presented below, colour coding will be used to indicate which actors have been given which emotion: blue actors are given the first emotion listed, red actors the second; black actors are neutral, with zero emotion values.

7.5.1.1 Strolling / Hurrying

[Figure 7.3, Video 7.1] This is one of the most obvious examples of all the seven pairs, due to the very different speeds of the actors: the *strolling* actors are walking extremely slowly, at the lowest end of the range of possible speeds, while the *hurrying* actors are walking as fast as possible. However, there are a number of other observable differences:

- *Hurrying* actors will never choose to slow down behind another actor – they will always try to overtake; this may involve waiting (and slowing down) until it is safe to do so, but in general they will thread their way through the crowds of slower moving actors. This leads to almost no lane formation among *hurrying* actors, but this does take place in *strolling* actors (top).

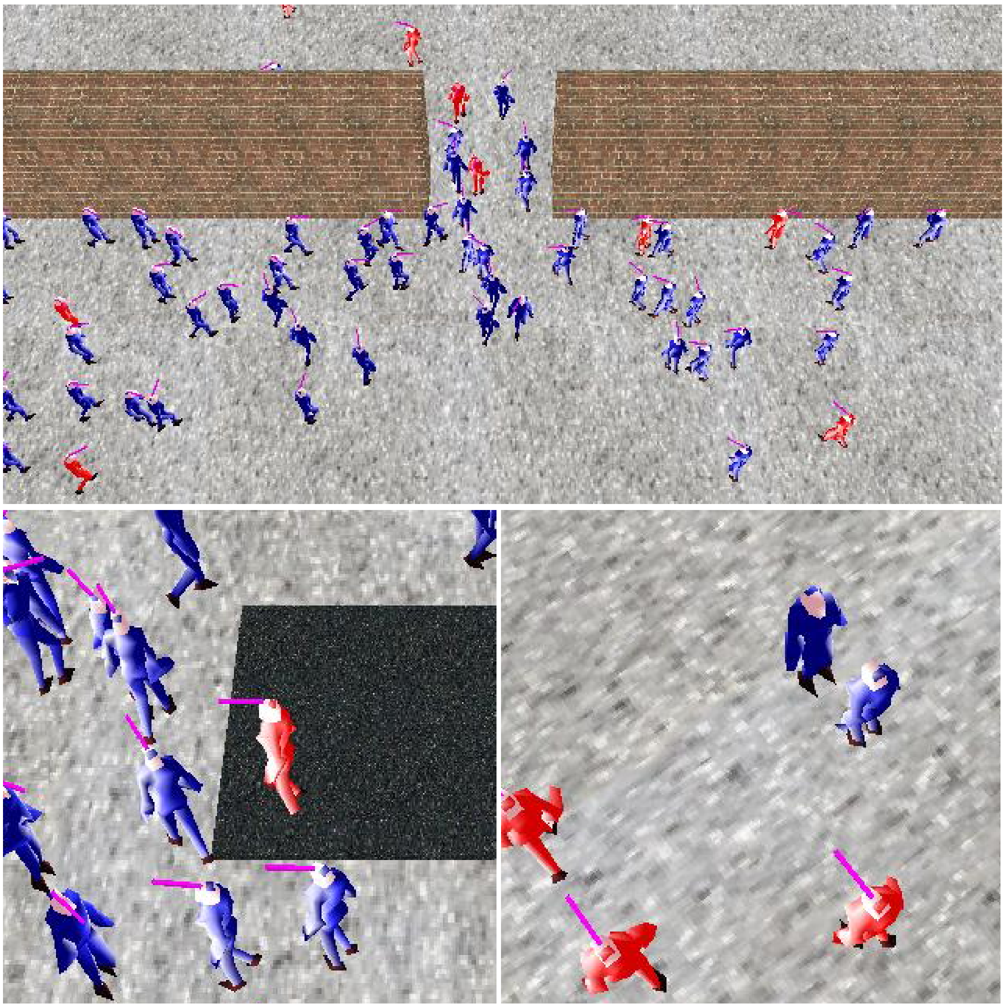

Figure 7.3: Blue: Strolling, Red: Hurrying

Figure 7.4: Blue: Carefree, Red: Afraid

- *Hurrying* actors will take risks: they can be seen to make use of the road when overtaking (bottom left).
- *Strolling* actors can be seen stopping to talk to each other (bottom right).

7.5.1.2 Carefree / Afraid

[Figure 7.4, Video 7.2] There is a clear difference in eye movements between the two types of actors: in particular, actors that are *afraid* can be seen looking behind themselves (when the purple gaze arrow points backwards – top). Their gaze also moves around more than *carefree* actors, as they inspect the whole scene for others that might be threatening them. They make larger detours around each other (bottom left), to avoid having to pass too close to others; *carefree* actors will pass much closer together (bottom right).

Figure 7.5: Blue: Excited, Red: Thinking

7.5.1.3 Excited / Thinking

[Figure 7.5, Video 7.3] The main difference in these emotions is visible in eye movements: *excited* actors look around far more (top) than those that are *thinking* – they look mainly straight ahead (bottom).

7.5.1.4 Interested / Bored

[Figure 7.6, Video 7.4] Effects on gaze are the same as for *Excited / Thinking*. An additional effect is that actors who are *interested* in those around them are more likely to make eye contact (visible as an extended period of time looking at another actor – left), and may also stop to talk to other actors of the same type (right).

Figure 7.6: Blue: Interested, Red: Bored [video only]

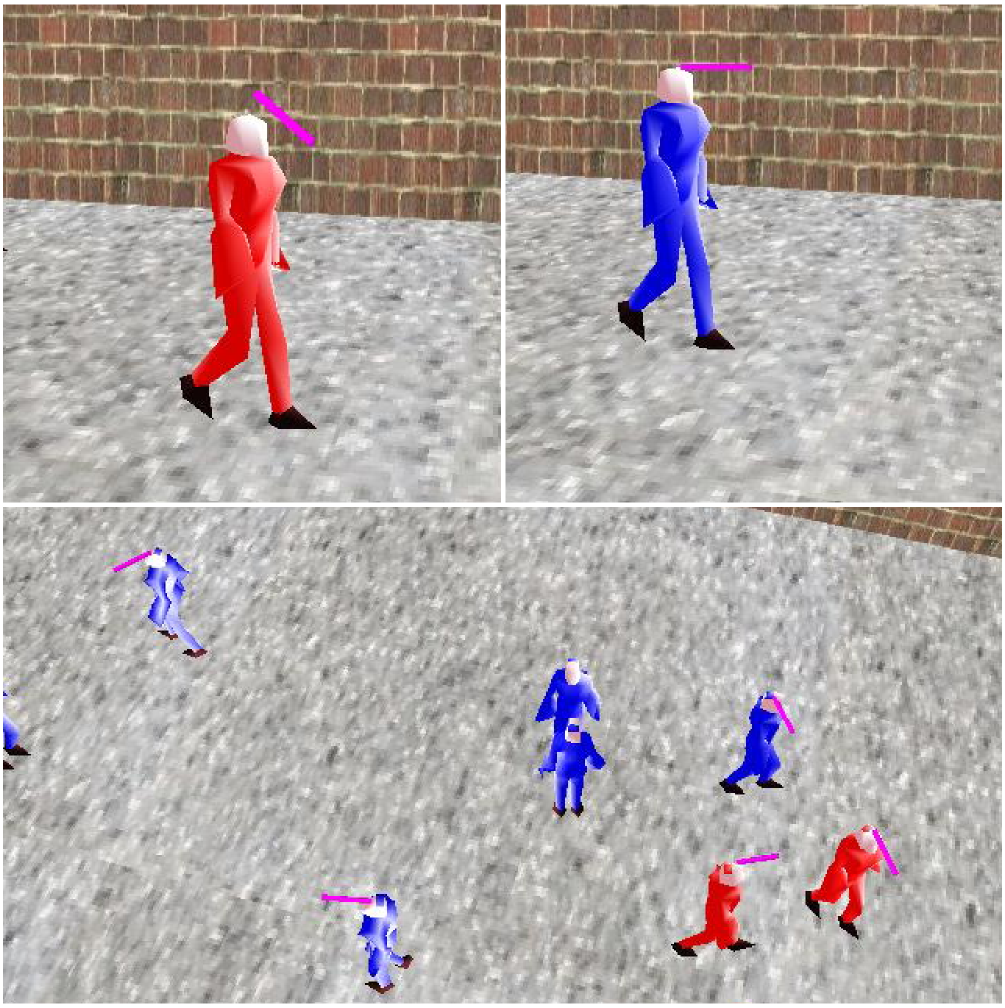

Figure 7.7: Blue: Happy, Red: Sad

Figure 7.8: Blue: Unfriendly, Red: Kind

7.5.1.5 Happy / Sad

[Figure 7.7, Video 7.5] The main visible difference is that *sad* actors look mainly straight ahead, and will tend to look down at the ground (top left) rather than up at other actors (as the *happy* actors do – top right). In addition, the *happy* actors can be seen stopping to talk to others (bottom).

7.5.1.6 Unfriendly / Kind

[Figure 7.8, Video 7.6] There are a number of clear differences visible here:

- *Unfriendly* actors will sometimes make no attempt at all to avoid a collision (right), leaving the other actor to do all the work. Even if they do participate, they will only detour late on, and by a small amount.
- *Kind* actors will always start to avoid a collision in plenty of time, and will leave a large amount of space around the other actor (left).
- *Kind* actors are continually making eye contact with others and frequently stop to talk to others (not shown).

7.5.1.7 Unsure / Sure

[Figure 7.9, Video 7.7] In terms of eye movements, *unsure* actors can be seen to spend far more time looking around to see where they are and who is around them (top). *Sure* actors tend to walk faster, are more likely to overtake – *unsure* actors prefer to form lanes – and will take risks by walking on the road (bottom).

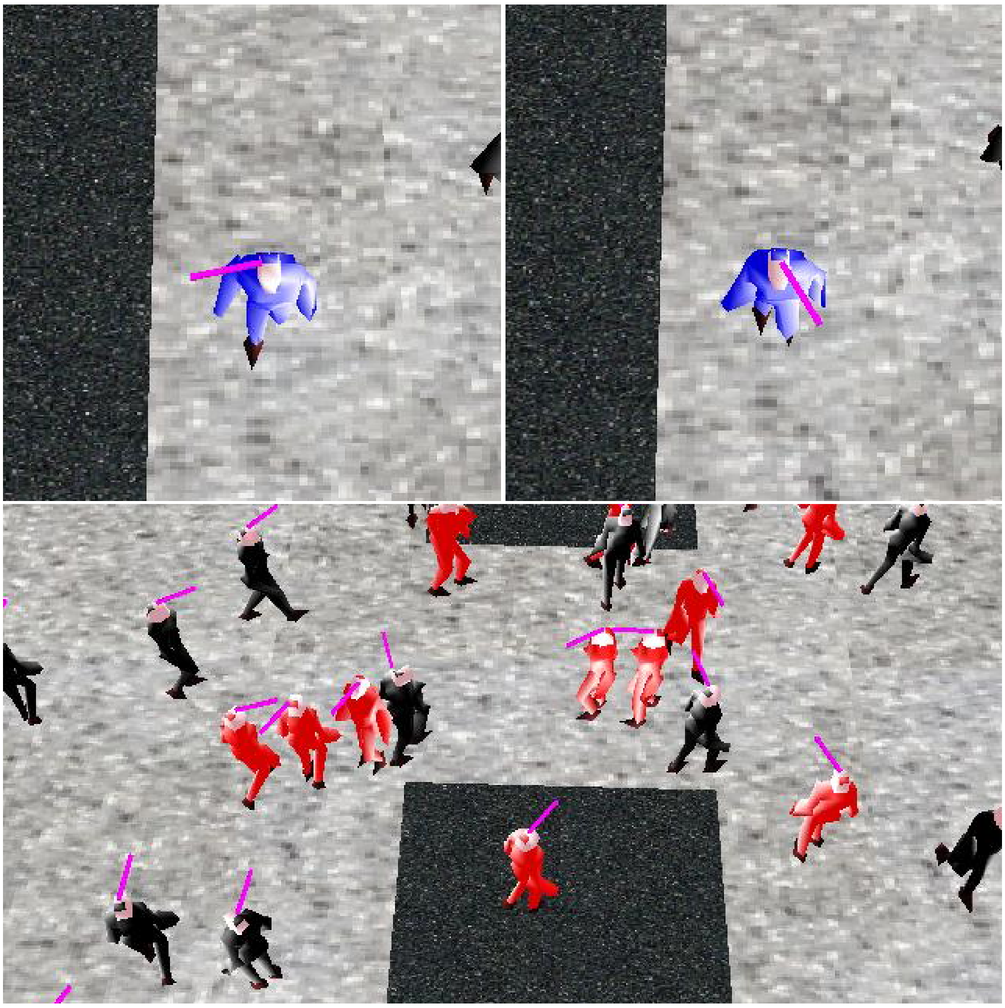

Figure 7.9: Blue: Unsure, Red: Sure

7.5.2 User testing

It was decided that a good way of evaluating the success of the simulation of moods and emotions was by performing user testing. Emotions cannot be evaluated quantitatively, as their interpretation is a matter of personal opinion: therefore, the only way of determining if emotions have been simulated correctly is if a majority of people can recognise what emotion is being shown.

The structure of user testing is discussed first, followed by the results obtained.

7.5.2.1 Test structure

The videos used for the test were those presented in the previous section: seven videos, each demonstrating two opposing emotions, such as Afraid and Carefree, with actors coloured red or blue to show which emotion they possess. It was decided to show the emotions in pairs as this enables easy comparison between opposite extremes, making the effect being demonstrated more obvious; for example, it is easier to tell that actors are walking quickly if others are walking slowly.

The videos each consisted of a series of clips, designed to highlight the unique features of its pair of emotions; for example, if an emotion caused actors to stop and talk to each other, several examples of this happening were shown. While these features would also be visible in a more general overview of a simulation, test participants would have had to watch for far longer to have a reasonable chance of noticing them. Therefore the decision to use a series of highlights enabled the test to be carried out in a reasonable amount of time.

The test was divided into two sections: identification of emotion pair, and determining which emotion was which.

1. Participants were given a list of the seven pairs of emotions in alphabetical order, and were shown the seven videos in a random order. They were instructed to write down the number of the relevant video next to each emotion pair. For each video the series of clips were first shown in sequence, and then repeated on a split-screen showing all the clips simultaneously; this was looped until all participants were happy to continue. After all videos had been shown, participants were allowed to watch any of them again.

2. Participants were shown the split-screen version of each video again; however, this time they were informed which pair of emotions it represented.

They were instructed to write down next to each emotion whether it was being shown by the red or blue actors.

A number of options were considered for the first part of the test: participants could have had the list of emotions withheld, and instead been given a free choice; alternatively, participants could have been given a shorter list to choose from for each video. The first option was dismissed on the grounds of making the test nearly impossible to evaluate; for example, would an answer of 'depressed' (which is not modelled) mean the same as 'sad'? The latter was considered to be too easy, and would have caused problems in deciding which emotions to include as choices. The chosen procedure was judged to be an appropriate balance between making the experiment too easy and too hard.

The second part of the test was felt to be necessary, in case participants found the first half difficult: if people were unable to recognise emotions, it was hoped that they would at least be able to tell opposing emotions apart when told the pair in question.

7.5.2.2 Test results

The test was conducted on twelve members of the Rainbow Research Group[1], most of whom had limited previous experience of human behaviour research. The raw results for the two parts of the test are shown in figures 7.10 and 7.11. The mean score in the first part was 4.7, and in the second part was 6.2 (both out of seven). The median scores were 5 and 6.5. A breakdown of the results by emotion pair is shown in figures 7.12 and 7.13. A confusion matrix for the first part of the test is given below; this shows the number of people that chose each emotion, compared with the correct answer.

		Chosen						
		StHu	SaHa	UK	AC	US	BI	TE
Correct	StHu	**11**						1
	SaHa		**8**					4
	UK		2	**7**	2		1	
	AC		1	1	**6**	3	1	
	US			2	2	**8**		
	BI			1	2		**9**	
	TE	1	1	1		1	1	**7**

These results are discussed in the next section.

[1]http://www.cl.cam.ac.uk/Research/Rainbow/

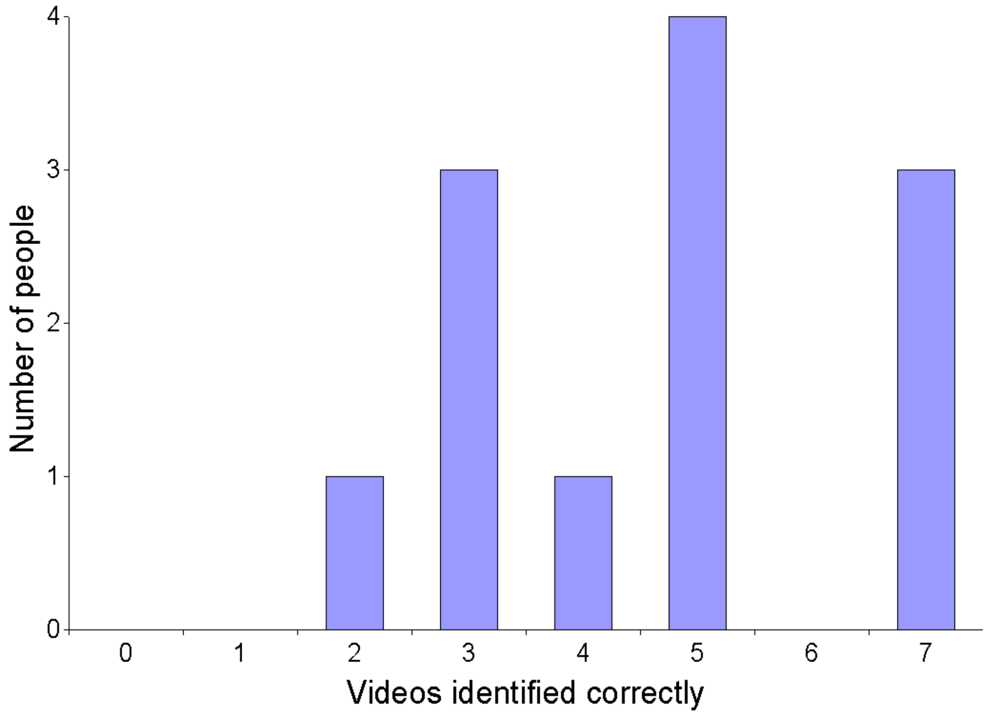

Figure 7.10: The raw results from part 1 of the test

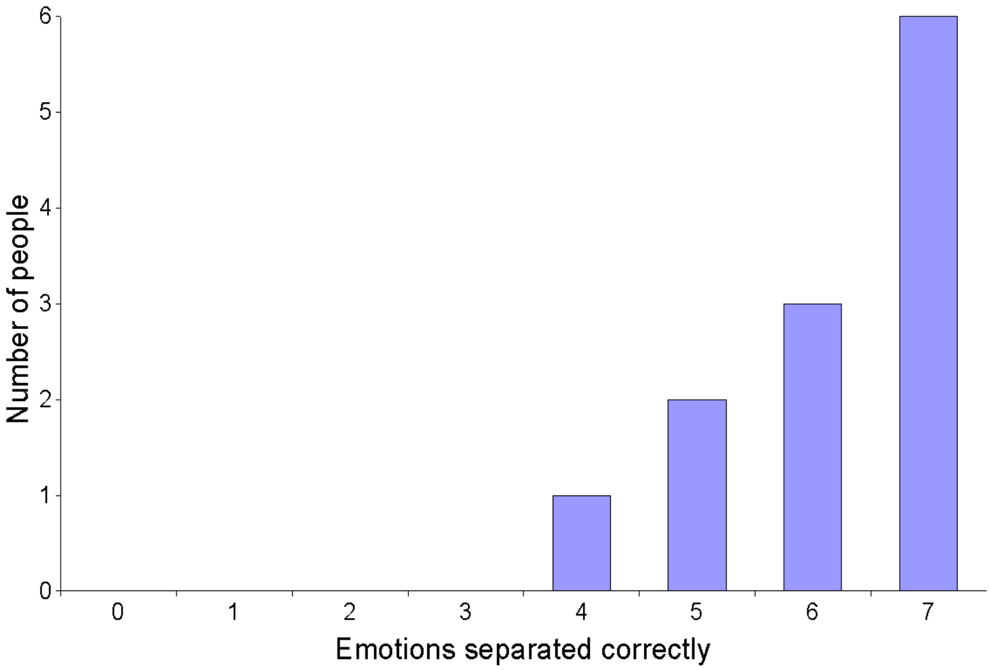

Figure 7.11: The raw results from part 2 of the test

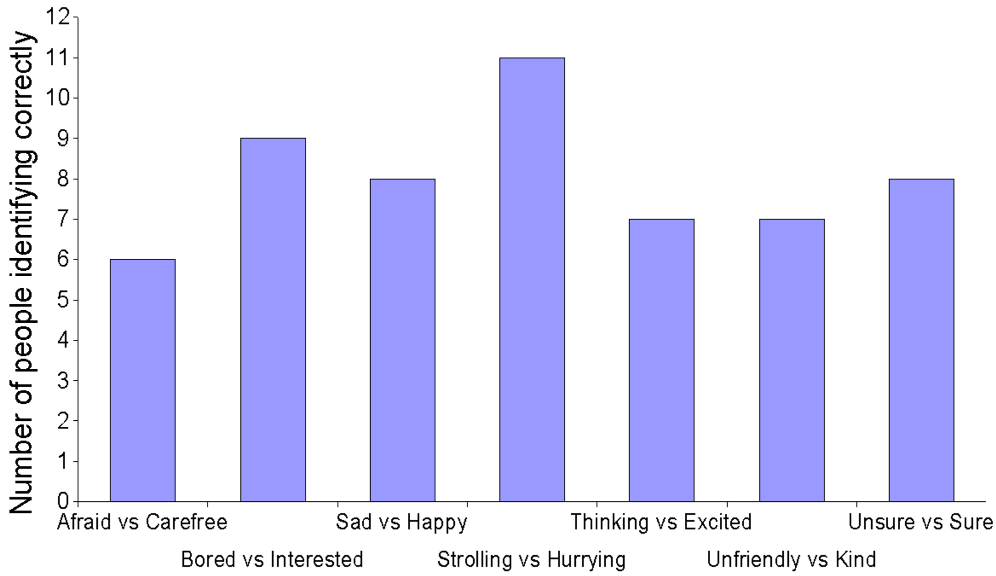

Figure 7.12: Results from part 1, analysed by emotion pair

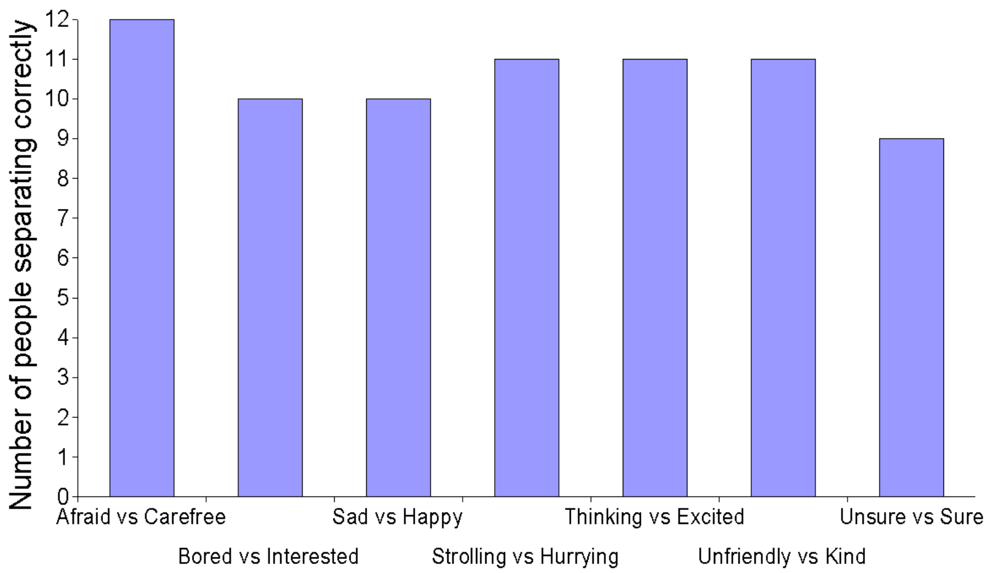

Figure 7.13: Results from part 2, analysed by emotion pair

7.6 Discussion

Following the presentation of graphical evidence and test results above, this section evaluates the success of this simulation of moods and emotions.

Graphical evidence The screenshots and videos in section 7.5.1 show that a variety of the effects of moods and emotions upon behaviour have been successfully simulated. Effects such as speed of walking, willingness to overtake, variations in eye gaze movement, and stopping to talk to people were all shown to occur when desired.

Test results The most noteworthy result from the video tests was that seven of the twelve test participants correctly identified five or more of the pairs of emotions; in other words, over half of the test participants were either entirely correct or made one mistake. Of the seven pairs of emotions, all were correctly identified by at least half of the test participants, and four by at least two-thirds of the participants.

The second part of the test proved even more successful: all but one of the emotion pairs were correctly isolated by ten or more people, and half the group achieved a perfect score.

Given the difficulty of the first task, having to choose from the complete list of emotions, these results show that the simulation was extremely successful: participants were indeed, on the whole, able to identify what emotions were being represented. The second task indicates that even people who could not originally identify an emotion pair were able to discern the effects upon behaviour when they had more idea of what they were looking for.

The confusion matrix shows that the only two pairs of emotions commonly confused were Afraid–Carefree and Unsure–Sure (shown in red); this is no doubt due to certain similarities in effects on behaviour, such as actors looking behind themselves. In addition Sad–Happy was fairly often identified as Thinking–Excited, but not vice versa; this could be due to participants identifying the looking down behaviour, and associating that with people thinking. The effect would not work in reverse, as Thinking actors are not shown looking down. Other than these examples the errors appear to be mainly specific to individual participants.

These results are also a clear validation of the proposal, discussed in section 3.3.2.3, that there are shared views about the effects of emotions upon behaviour: if there were not, this experiment would have been unsuccessful, as people would identify effects on behaviour in different ways.

Conclusion The evidence shows that this simulation has created a believable representation of moods and emotions which observers can identify. One of the main uses for this type of simulation is creating virtual actors for a film with which a viewer can empathise, so this is the best way of judging the system: as the participants in the test were able to determine the emotion being shown, they would indeed be able to empathise with those actors. It is also noteworthy that the results are good given the low realism of the way the actors are actually drawn.

Clearly in a more fully-featured simulation there would be other ways of demonstrating the emotional state of an actor, for example, its facial expression, body language and dialogue; this would make the task of identifying the emotion easier. Therefore, the fact that emotion is discernible simply from the effects on walking behaviour and eye-gaze movement is an indication of how successful a full simulation, combining all of these different aspects, could be.

7.7 Summary

This chapter has explained the way in which a simulation of moods and emotions has been added to the system described in previous chapters. The links between moods and variations in behaviour were described, and each variation was covered in detail, along with implementation techniques.

Graphical evidence of the simulation was then presented, followed by the results of an experiment which demonstrated that people could identify different emotions from their effects on behaviour, showing that this work has been successful.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The aim of this work was to demonstrate that using ideas from psychology research would lead to a simulation of human behaviour that was more accurate and more realistic than previous work. This section describes the ways in which this has been achieved, looking at each main area of research in turn: collision avoidance behaviour, vision and attention, and moods and emotions.

8.1.1 Collision avoidance behaviour

This work incorporates a number of novel aspects of behaviour that have been taken directly from psychology research:

- Actors can perform the step-and-slide manoeuvre as an alternative to detouring, or in combination with it. It is an extremely common feature of everyday behaviour, and allowing actors the option of performing it adds to the variability as well as the realism of the simulation.

- The technique used to determine the side on which actors should pass each other when avoiding collisions matches that used by real people, rather than simply being a random choice.

- Actors have the option of slowing down to avoid collisions rather than overtaking; this leads to the emergent behaviour of lane formation, a commonly observed real-world phenomenon.

- Actors will determine if there is room to overtake before trying to do so, and will slow down while they wait.

None of these has been properly simulated in the past. Not only do they make the simulation look more realistic, in that people will recognise them as re-creating common human behaviour, but they also improve the accuracy, meaning that experiments such as those determining evacuation times produce results closer to real life.

8.1.2 Vision and attention

This work is the first to include a complete, psychology-based model of attention as it applies to the detection and monitoring of other actors. This improves the simulation in a number of ways:

- The eye gaze movements of each actor are simulated: an observer can see what an actor is interested in, and which actors it is tracking. This makes the actor look far more like a real human than if it were simply staring forwards.
- Each actor builds up its own mental model of the scene, and is allowed to use only this to determine its behaviour. This leads to a 60% increase in the number of collisions, re-creating the real-world effect of actors colliding due to being distracted and not realising what would happen until it is too late.
- The overall speed of movement of actors is slower than if attention were not simulated, due to the increased number of collisions and near misses, where the actors have to slow down. This has a significant effect upon the time taken to evacuate, which was found to increase by around 50% in one example.

Attention simulation therefore not only causes an improvement in realism, as actors have believable eye gaze movements, but also produces a more accurate simulation; this allows better results to be obtained, such as in the comparison of different building layouts. It also leads to more natural, less artificial, behaviour.

8.1.3 Moods and emotions

Previous work has considered the effects of emotion on body language, but this work is the first to simulate the effects that the moods and emotions of actors have upon their walking behaviour and upon their eye-gaze movements. This leads to the following improvements in the simulation:

- There is a great deal of variability in the behaviour of the actors in the simulation: some will look around a lot, others will stare straight ahead, while some will look down at the ground; some will appear to be in a hurry, and thread their way through a crowd, while others are content to go with the flow; some actors will walk confidently over the road, while others will fearfully look behind themselves. All these features lead to a far more interesting simulation than if all the actors behaved in the same way.

- User testing proved that observers are able to identify the mental state of actors: based solely on variations in behaviour, they could tell what mood was being represented with a high degree of accuracy – around 70%.

If virtual actors are to be used in films, it is essential that they are able to represent different emotions, in the same way that real actors do, so that viewers can identify with them. This work has showed that this is indeed possible, and has demonstrated ways in which it may be successfully achieved.

8.1.4 Overall

The evidence just presented shows that the original hypothesis behind this work has been proved correct – making use of psychology research has led to a more accurate simulation of human behaviour:

- Actors' behaviour contains more features from the real world.
- The simulation of attention leads to more natural behaviour.
- Actors can be given recognisable moods and emotions.

Therefore, the real contribution of this work is showing that psychology research is a rich resource that can be used to improve behaviour simulations. Rather than tweaking ad hoc rules until they give a satisfactory result, making use of ideas from psychology at all stages of the simulation process will lead to accurate and realistic results. The system shown here is an example of this approach, and its success demonstrates the validity of the idea.

8.2 Future Work

A number of areas in which further work could be carried out have been identified.

8.2.1 Hearing

In this work the only human sense that has been considered is vision: all the information about the positions and velocities of other actors comes from what has been seen through shifts in eye gaze. However, in the real world people also acquire some relevant information through their sense of hearing. Clearly it is not normally possible to determine the exact whereabouts of another person just by listening, but there is one particular case when audition can prove very useful: discovering that a person is approaching from behind. People do not generally look behind themselves very often, unless they are particularly afraid, so these situations would not be detected simply using vision. Audition only provides one piece of information: that there is somebody behind; exact information would only be determined when the person turned his head to look behind.

This feature of human perception could be implemented in one of two ways:

- A simple approach would be to trigger an Explicit Actor Request when any actor approaching from behind the subject got within a certain distance; the subject would then turn round to look at it. Clearly this is not in any real sense a simulation of audition, but it would cause the same effects.
- The more complicated approach would be to simulate the production of noise by each actor, mainly from its footsteps. The inverse square law would then be used to determine how the sound propagated through the world, until it reached the subject. If the sound were above a certain volume, implying that the actor in question was nearby, then the subject should be instructed to turn and look. Sounds clearly might be blocked by walls, and by the presence of other actors, so occlusion would have to be considered. An approach similar to this is used by Massive [Mas03].

Given the current prevalence of personal music players it could be argued that the sense of hearing is becoming less of a factor in people's awareness of their surroundings, and that this system, without audition, is therefore a better approximation of reality.

8.2.2 Obstacle avoidance

The obstacle avoidance procedures used in this system are fairly simple, as they only apply to walls or other large, stationary objects: actors will not be given paths that would take them through obstacles, avoidance procedures are not chosen if they consist of walking through an obstacle, and, as a last resort, an actor will be stopped from colliding with an obstacle by putting it back on a safe route. The last of these, although it is rarely needed, can look unnatural, as the actor may walk right up to a wall before turning back, sometimes requiring a sudden change in direction.

There are several ways in which this behaviour could be improved:

- An actor could check at all times whether it will collide with an obstacle in the near future, and correct its course if required.
- Actors should be given a way to avoid small stationary obstacles that are not part of the route map, such as banana skins on the ground; these require small corrections as the actor walks, rather than planning in advance, similar to the avoidance of other actors. This was considered in previous work by Gillies [Gil01].
- Actors could be given extra rules to deal with moving obstacles, such as cars – also explored by Gillies. Actors can be given choices: speed up to pass the car before it arrives, or slow down and wait until it passes, both of which may be combined with lateral movement.

8.2.3 Route planning

The actors in this system use A* search to determine the shortest route through the scene to get to their goal; while doing this they have complete access to the map of the scene. This re-creates the situation where people are very familiar with their environment, and can therefore accurately plan the best route to take. However, people are often not so knowledgeable about their surroundings: if they are visiting an area or a building for the first time then they may have to explore to find out how to get to their destination.

The topic of navigation through unfamiliar environments has been studied in depth – a brief overview was given in section 2.2.3.3. The most useful type of route planning would appear to be 'Survey Navigation', in which an actor has to build up its own model of the scene as it walks around; it can then make use of this information in future route planning.

The simple approach to implementing this type of navigation would involve the actor walking around fairly randomly, slowly building up its model, until it was able to discover its goal. However, this does not reflect a common way that real people discover how to get somewhere: they will follow signs. Depending on the location the number of signs will vary: an underground station would be full of them, while a shopping street might not have many; importantly, fire exits are normally marked extremely well in buildings, and being able to recognise these would be particularly useful in an evacuation simulation.

There is also scope for route planning to be affected by the moods and emotions of the actor: some might choose to take the shortest possible route to their goal if they are in a hurry, while others may prefer to take a more scenic route. In addition, actors who are thinking, and not concentrating on their route planning, could end up getting lost.

8.2.4 Goals

Each actor in this work is given a series of goals, which are map squares chosen fairly randomly. This is sufficient for the purpose of giving the actors a reason to walk around and in the process cause potential collisions – the real area of interest of this simulation. It is also all that is required for two of the main uses of this type of system: in films, effects shots are normally extremely brief, so there will be no time to see actors reaching their goals, and in evacuation simulations the only goal is to escape from the building – as long as it is outside the building the exact map square chosen is unimportant.

However, there are some occasions when actors' long-term behaviour might become important, perhaps during a long scene in a film where virtual actors were walking in the background. Therefore, it would be worthwhile to be able to give useful goals to actors, for example:

- Actors in a shopping centre could be given goals to walk around and enter shops.
- Actors in a park could be instructed to stroll around, run in circuits, or lie on the grass.
- Actors in a railway station could be given goals of queueing up to buy tickets and walking to a platform.

In other words, the choice of goal should be based on some mental state of the actor, and on its situation. The user of the system should be able to specify high-level goals (such as 'visit shops'), and suitable goals would then be generated automatically. The easiest way of implementing this would be to program the high-level goals explicitly into the system, each with a set of rules for producing goals. There has been previous work in this area, for example by Shao and Terzopoulos [ST05].

For this method to work, the map of the scene would need to contain more information: the system would need to know which parts of the map were suitable targets, such as shops.

8.2.5 Groups

All the actors in this simulation walk around on their own, except for effects caused by lane and cluster formation. However, real people often walk around in couples, or in larger groups of friends, depending on the situation. For the purposes of evacuation simulation single people are probably sufficient: people are less likely to be in groups in these types of situation, particularly if the location is, for example, an office building. In a street scene, though, the believability of the simulation would be increased by some people walking in pairs.

The rules describing the behaviour of couples were examined by Goffman [Gof72] and Wolff [Wol73]. Goffman calls groups of people 'withs', and notes that the most common type of with consists of two people; their defining characteristic is that they maintain sufficient proximity to allow easy conversation and to ensure exclusion of others. Wolff further explains that "people walking in couples, especially when engaged in conversation, will rarely separate or yield to an oncoming single pedestrian... [They will] expect that the oncoming pedestrian will cooperate by detouring around them, allowing them to continue without interruption." From this several rules for the simulation of pairs are clear:

- Both members of the pair will have the same goal, and will be following the same set of sub-goals to get there.
- The two actors will try to keep the distance between them constant.
- They will be more unwilling to avoid Towards collisions; if they have to, they will prefer to move together to one side of the collidee, rather than separating.
- They will be more likely to slow down to avoid an Away collision; if they have to overtake, they should try to do so on the same side, and may briefly form a two-person lane while doing so.
- If they do need to separate to avoid a collision, they should come back together as soon as possible afterwards.
- Single people will need to avoid the pair as a single entity – this is already handled in this system (section 5.2.1).

8.2.6 Eye contact

The model of eye contact used in this system (section 7.3.1.5) is a simplification of reality; it makes actors more likely to look at each other if they are friendly, and less so if they are not. A more advanced system would need to consider the difference between looking at someone, and making eye contact with them: the latter specifically implies that *both* actors are looking at each other. However, people are less likely to look at someone if that person is currently looking at them: if eye contact is made, both people may look away rapidly, as if to pretend that they had not been looking at the other in the first place.

There are clearly differences in eye contact behaviour depending on the relationship between the two people. If they know each other then they may be willing to make eye contact as a form of greeting, perhaps accompanied by a smile. On the other hand, strangers are far less likely to allow eye contact. A simple way of simulating this would be to randomly decide whether two actors 'know' each other, and allowing or forbidding eye contact based on this.

8.2.7 Graphical display

The 3D graphics display was designed simply to show what was happening in the simulation, and not to look particularly realistic. It met this requirement, as observers were able to determine what behaviours the actors were performing during the moods and emotions user testing (section 7.5.2).

However, it would clearly be an advantage if the graphics could be improved, as this would make the simulation more enjoyable to watch, and might also help with understanding of behaviour. This could be achieved in two ways:

- Try to improve the current graphical system by using more accurate models of people, animation based on motion capture data, better texturing, and proper lighting.

- Implement the capability to export position data for each actor on every frame of the simulation, and import this into a commercial rendering system, such as 3D Studio Max[1] or Maya[2].

The first option has the advantage of keeping the simulation interactive, allowing the user to test different settings quickly; however, a great deal of work would be needed to produce a realistic system, and it would, in a sense, be 'reinventing the wheel'. The second option would produce better-looking

[1] www.autodesk.com/3dsmax
[2] www.autodesk.com/maya

graphics for less work, but would mean that any simulation had to be fully rendered before it could be watched – if an error was made then the whole sequence would have to be redone.

The best solution to the problem would seem to be a small improvement in the current graphical system – such as using more realistic models – combined with the ability to export to a rendering system. The real-time interactive display would then be used to get the simulation set up correctly, before being fully rendered for its final use.

8.2.8 Camera control

The control of the camera in this system is left entirely to the user: it is possible for the user to select any viewpoint and look direction, or a variety of tracking options. An extension to this would be a system that would automatically focus the camera view on an area where something interesting was happening, for example, an area which is becoming congested due to a number of collisions taking place. In addition, if a particular actor is being followed then the system could switch between over-the-shoulder, top down, and long distance tracking shots, moving the position of the camera to ensure there is always a good view of the actor.

This type of system would probably not be very useful in films, as the director would want to make this sort of decision himself; however, it could prove useful in evacuation simulations, where ensuring visibility of behaviour is more important than an artistic composition.

8.2.9 Summary

The preceding sections have shown that there is scope for a large amount of extra work in the field of simulation of human behaviour: there is still a long way to go before virtual actors are indistinguishable from real people.

Appendix A

Programming details

A.1 Detecting actual collisions

In section 5.2.6.2 a function was defined which would return details of any actors which were actually overlapping with the subject.

The process begins by considering each actor as a circle, and checking how far apart their centres are: if the distance between them is greater than twice the radius of a circle then no collision is possible and there is no need to perform any more expensive tests.

Circles are not an accurate representation of the shape of an actor. Therefore, more detailed checks are conducted where an actor is considered to be an ellipse, designed to fit over the width of its shoulders. To detect if two actors are overlapping it is necessary to check for an intersection of ellipses. A rigorous mathematical intersection would be algebraically complicated and not justified by the minor extra precision it would give. Instead, eight points are chosen around the edge of the ellipse representing one actor, and each of these is checked to see whether it is inside the ellipse of the other actor; the eight points are shown by the black dots in figure A.1. This process is then repeated with the actors swapped; this detects those cases where one of the eight points on one actor is inside the other, but the opposite is not true.

If an actor is performing a step-and-slide then the ellipse representing it is rotated through 90°, and positioned at one end of the actor – shown by the red half-ellipse and red dots in figure A.1: only the front half of the ellipse is considered in the tests.

For each of the eight points around the perimeter of the ellipse, its coordinates are found in world space – rotated from the object space of the actor. Those coordinates are then rotated again to fit into the object space of the secondary actor, before they are checked to see if they are inside its ellipse.

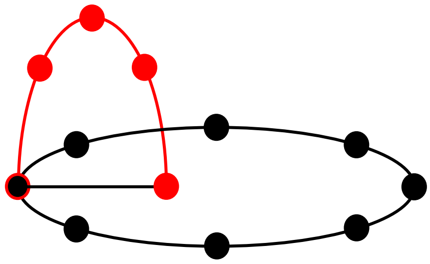

Figure A.1: The points around an ellipse used to test for overlap in normal conditions (black) and during a step-and-slide (red)

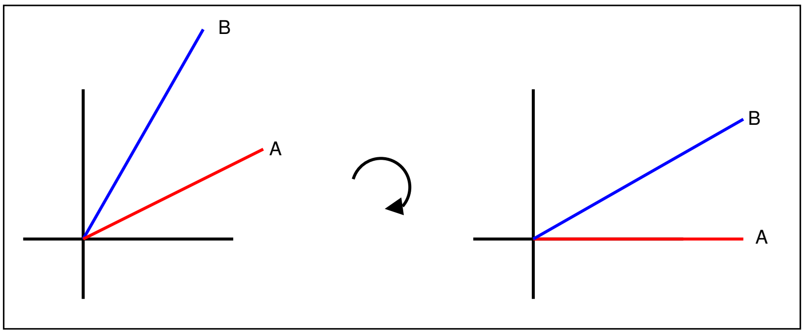

Figure A.2: Determining which side a vector is on

As soon as an intersecting point is found the function returns with details of the collidee, which is added to a stack kept by the subject.

A.2 Determining sides

It was found that it was frequently necessary to determine whether a vector was to the left or right of another vector. For example, in figure A.2, vector B is to the left of vector A.

The idea behind the function is to rotate both vectors so that vector A is pointing along either the positive or negative x-axis; to this end, the function finds the angle between vector A and the x-axis. Vector B is then rotated by this amount, clockwise if vector A is in the first or third quadrants, otherwise anti-clockwise. There is no need to actually rotate vector A, and in fact only the new y co-ordinate of vector B is needed. If vector A would now be pointing along the negative x-axis then the sign of vector B's y co-ordinate is flipped.

Finally, if vector B's y co-ordinate is positive it is to the left of vector A, if negative it is to the right.

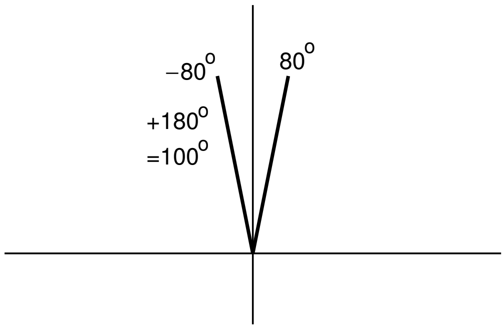

Figure A.3: Vector angle manipulation

A.3 Vector angles

There is a slight subtlety to the calculation of direction vector angles, due to the discontinuity of the tangent function used to perform the calculation. This has the effect that if, say, the first vector was in the first quadrant, and the second vector in the second quadrant, they would appear to be at very different angles, when in fact they are close together. For example, in figure A.3 the two vectors have a difference in value of 160°, when they are in fact only 20° apart.

Therefore, a check is made to see if the two vectors are on opposite sides of the y-axis. If they are, with one in the first quadrant and one in the second, 180° is added to the angle of the vector in the second quadrant; or, if one is in the third quadrant and one is in the fourth quadrant, 180° is added to the angle of the vector in the fourth quadrant. Any other cases with the vectors on opposite sides of the y-axis can be ignored, as the vectors will be in opposite quadrants, and are therefore not interesting for any of the purposes for which this algorithm is used.

A.4 Selecting actors

Users of the system can click on the 3D display and the nearest actor to that location will be selected, as described in section 6.1.3.1. The technique for this calculation is as follows:

1. Windows returns the coordinates on the 2D pictureBox (used to display the DirectX scene) where the click happened.

2. These coordinates are converted to *viewing coordinates* – i.e. the coordinates of that point on the 2D screen onto which all of the 3D world is projected, before then being scaled and displayed on the pictureBox. This conversion is performed using the ratio of the size of the pictureBox to

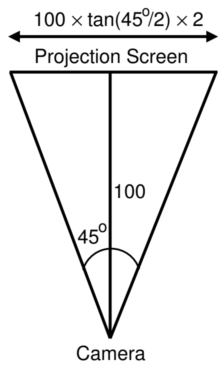

Figure A.4: Calculating the size of the projection screen

the size of the projection screen. The latter is found using the distance from the camera to the centre of the screen (100) and the field of view (45°), giving the width of the screen as $100 \times \tan(\frac{45^\circ}{2}) \times 2$; this is shown in figure A.4. The position of the click in viewing coordinates is found by multiplying the coordinates of its position on the pictureBox by this ratio.

3. The position of each of the actors in the scene in viewing coordinates is found, by multiplying its position in world coordinates by the viewing transform matrix used by DirectX.

4. The magnitude of the cross product of a vector representing the position of the actor, and the direction vector of a line from the origin to the point of the click (in viewing coordinates), gives the minimum distance between the line and the actor.

5. The nearest actor to the clicked position is selected.

A.5 Mood-behaviour calculations

Mood-behaviour calculations consist of a one-off initialisation of the mood-behaviour table (section 7.2.2), followed by processing each actor in turn.

A.5.1 Mood-behaviour table initialisation

Once the mood-behaviour table has been filled with values (section 7.2.2.2), either from stored data or through the user interface, the system calculates the maximum and minimum possible values for each of the behaviour-variations, by adding either the maximum or minimum value of each mood range for that behaviour-variation. That is:

$$\begin{aligned} \text{minimum} &= \sum_{v_j<0} v_j \\ \text{maximum} &= \sum_{v_j>0} v_j \end{aligned}$$

where v_j are the values of one row of the mood-behaviour table for the behaviour-variation in question. For example, in the mood-behaviour table shown in figure 7.1, the minimum value for 'Do not avoid' is $(-100)+(-100) = -200$ and the maximum value is $25 + 100 = 125$.

This information is required to show the degree of effect that each of the moods has upon a behaviour-variation; for example, if the maximum value for a behaviour-variation was 200, then a value of 100 for a particular mood means that the setting of that mood will affect 50% of the value of the behaviour-variation.

A.5.2 Actor behaviour setup

To calculate each of the twelve behaviour-variation values for an actor the following procedure is used (an example follows):

1. Start a running total, and consider each mood value.
   (a) If it is less than zero, it is towards the left-hand end of the range (e.g. 'sad'); its absolute value is multiplied by the entry in the mood-behaviour table for that mood (here 'sad') divided by 100, and added to the total.
   (b) If it is greater than (or equal to) zero, it is towards the right-hand end of the range (e.g. 'happy'), and is multiplied by that entry from the mood-behaviour table divided by 100, and added to the total.
2. The sign of the total is considered.
   (a) If it is less than zero, it is multiplied by -100, and divided by the minimum value for the behaviour found in section A.5.1.
   (b) If it is greater than (or equal to) zero, it is multiplied by 100, and divided by the maximum value.
3. This value is stored as the value for the behaviour-variation being considered for the current actor.

Example In the mood-behaviour table shown in figure 7.1 the 'Do not avoid' behaviour can be seen to be affected by three pairs of emotions. The maximum value is 125, and the minimum -200. Suppose the Strolling-Hurrying mood has been set to 100, and all the others to zero; the calculation will then be:

$$\left(\frac{100 \times 25}{100} + 0 + ... + 0\right) \times \frac{100}{125} = 20 \tag{A.1}$$

If the value of the mood Unfriendly-Kind is now altered between 0 and -100 (making the actor more Unfriendly) the behaviour value will vary between 20 and 100; for example, with a mood value of -100:

$$\left(\frac{100 \times 25}{100} + \frac{|-100| \times 100}{100} + ... + 0\right) * \frac{100}{125} = 100 \tag{A.2}$$

In other words, the effect of Unfriendly is four times greater than the effect of Hurrying; this is due to their values in the mood-behaviour table of 100 and 25.

If Afraid-Carefree alone is set to -100 (making the actor afraid), then the calculation is:

$$\left(\frac{|-100| \times -100}{100} + 0 + ... + 0\right) \times \frac{-100}{-200} = -50 \tag{A.3}$$

It can be seen that 'Afraid' has a 50% effect on the behaviour, due to its value of -100 out of a total in the mood-behaviour table of -200.

Appendix B

Numerical results

B.1 Number of collisions

Raw results for the experiment described in section 6.2.2.1 [Number of collisions]: the tables show the number of collisions occurring in each of three modes of vision: total awareness, attention simulation and no awareness. Five sets of starting positions for actors were used, with ten runs of the simulation for each. Average values are shown for each set of starting positions, along with percentage increases compared to total awareness. Testing was performed on two scenes: an evacuation scene, and a typical street scene.

B.1.1 Evacuation scene

Total awareness	Attention	No awareness
17	34	44
21	18	46
21	38	44
15	34	46
19	42	49
25	30	45
18	32	43
21	40	38
13	26	44
21	28	40
Average: 19.1	32.2	43.9
	+68.6%	+129.8%
15	23	46
18	25	43

17	28	38
18	24	36
19	23	38
14	26	46
15	25	42
9	22	33
15	30	41
17	29	48
Average: 15.7	25.5	41.1
	+62.4%	+161.8%
17	31	47
21	27	37
18	27	42
25	35	40
21	27	37
20	33	44
19	32	39
21	34	35
20	31	40
21	32	41
Average: 20.3	30.9	40.2
	+52.2%	+98.0%
22	30	38
18	25	41
19	31	45
17	35	32
8	26	36
19	35	32
23	22	34
16	22	39
14	25	36
16	33	36
Average: 17.2	28.4	36.9
	+65.1%	+114.5%
15	27	44
21	28	40
17	30	40
12	24	37

19	29	42
20	24	35
16	30	38
22	34	42
13	23	37
17	33	35
Average: 17.2	28.2	39.0
	+64.0%	+126.7%

Average percentage increase over all five sets of starting positions compared to total awareness: 62.5% for attention, 126.2% for no awareness.

B.1.2 Street scene

Total awareness	Attention	No awareness
20	39	57
26	35	66
32	24	60
21	44	65
21	38	65
25	43	65
19	40	62
32	37	58
26	43	64
21	40	58
Average: 24.3	38.3	62
	+57.6%	+155.1%
25	42	52
34	39	58
29	43	64
17	36	59
30	38	63
24	41	56
31	42	59
28	39	63
23	37	60
20	44	61
Average: 26.1	40.1	59.5

	+53.6%	+128.0%
31	41	60
16	46	61
31	46	61
30	44	57
24	37	61
24	43	63
23	44	61
31	45	63
34	43	61
23	43	63
Average: 26.7	43.2	61.1
	+61.8%	+128.8%
26	43	59
16	48	61
31	36	62
30	39	63
36	33	62
18	45	58
29	43	67
24	43	60
27	43	65
28	33	62
Average: 26.5	40.6	61.9
	+53.2%	+133.6%
24	38	62
22	29	62
22	38	61
21	33	61
15	23	63
21	36	66
24	35	64
27	33	63
25	40	61
21	35	60
Average: 22.2	34	62.3
	+53.2%	+180.6%

Average percentage increase over all five sets of starting positions, compared to total awareness: 55.9% for attention, 145.2% for no awareness.

B.2 Evacuation time

Raw results for the experiment described in section 6.2.2.1 [Time to evacuate]: this table shows the number of frames taken for 100 actors to evacuate from a building. Five runs were carried out for each of five sets of starting positions for actors, using total awareness and attention simulation. Averages are given for each set of starting positions, along with the percentage increase for attention simulation compared to total awareness.

Total awareness	Attention
338	469
313	429
337	475
326	483
330	662
Average: 328.8	503.6
	+53.2%
328	499
325	468
336	443
335	516
328	545
Average: 330.4	494.2
	+49.6%
330	541
352	549
337	583
344	628
346	421
Average: 341.8	544.4
	+59.3%
371	689
398	600
407	572
357	450

385	646
Average: 383.6	591.4
	+54.2%
362	565
368	673
377	524
376	519
359	576
Average: 368.4	571.4
	+55.1%

Average percentage increase in evacuation time using attention simulation compared to total awareness: 54.3%.

B.3 Effects of distractions

Raw results for the experiment described in section 6.2.2.2: this table shows the number of collisions occurring with and without the presence of a distraction. Five runs were carried out for each of five sets of starting positions for actors; averages are shown for each set of starting positions, along with the percentage increase when a distraction is present.

No distraction	Distraction
26	37
28	33
24	33
30	29
16	35
Average: 24.8	33.4
	+34.7%
28	33
25	22
32	30
27	33
31	35
Average: 28.6	30.6
	+7.0%
21	22

20	28
18	26
23	29
18	32
Average: 20.0	27.4
	+37.0%
25	33
27	25
26	25
30	27
16	32
Average: 24.8	28.4
	+14.5%
25	30
12	34
24	27
15	27
24	20
Average: 20.0	27.6
	+38.0%

Average percentage increase in number of collisions when a distraction is present: 26.2%.

B.4 Testing building layouts

Raw results for the experiment described in section 6.2.2.3: this table shows the number of frames taken for 100 actors to evacuate a building using two different door layouts. Ten sets of starting positions for actors were used, one for each run. The average times to evacuate are shown, along with the percentage decrease in time when using two exits.

1 exit	2 exits
382	367
534	306
373	330
440	340
340	311
400	344

374	360
385	331
374	312
392	304
Average: 399.4	330.5
	−17.3%

Average percentage increase in speed of evacuation with 2 exits: 17.3%.

B.5 Attention shifts

Two examples of the attention shifts of an actor, as described in section 4.3.2.2 [Equation testing]: the tables show all the attention shifts of one actor, with one shift per row: 'Target' is the *id* of the actor being looked at, 'Type' shows the type of observation, 'TTC' shows the Time To Collision with the chosen target (if relevant), and 'Overdue' shows how overdue the chosen target was for a monitoring observation (with a negative value showing how long until it would become overdue).

Target	Type	TTC	Overdue
53	Periphery		
70	Spontaneous		
66	Spontaneous		
73	Spontaneous		
56	Spontaneous		
65	Spontaneous		
53	Monitoring		0
70	Monitoring		0
66	Monitoring		0
56	Monitoring	33	1
56	Monitoring	30	-7
56	Monitoring	28	-8
15	Spontaneous		
7	Spontaneous		
33	Request		
53	Request		
7	Monitoring	28	-23
7	Monitoring	28	-26
53	Request		

7	Monitoring	24	-24
7	Monitoring	23	-26
56	Periphery		
70	Monitoring		2
73	Monitoring		5
52	Periphery		
52	Monitoring	6	-4
52	Monitoring	6	-4
52	Monitoring	5	-3
73	Monitoring		1
38	Periphery		
38	Monitoring		0
73	Spontaneous		
26	Spontaneous		
22	Periphery		
9	Spontaneous		
7	Spontaneous		
7	Monitoring	25	-26
7	Monitoring	27	-26
49	Periphery		
7	Monitoring	33	-26
22	Monitoring		14
7	Monitoring	35	-26
7	Monitoring	37	-26
34	Request		
7	Monitoring	40	-26
7	Monitoring	42	-26
22	Monitoring		6
9	Monitoring		5
7	Monitoring	46	-26
7	Monitoring	49	-26
67	Spontaneous		
35	Spontaneous		
47	Spontaneous		
10	Spontaneous		
72	Spontaneous		
49	Monitoring		0
22	Monitoring		1

69	Spontaneous		
54	Spontaneous		
16	Spontaneous		
24	Spontaneous		
22	Monitoring		1
67	Monitoring		1

Target	Type	TTC	Overdue
12	Periphery		
66	Spontaneous		
38	Spontaneous		
48	Spontaneous		
37	Spontaneous		
58	Monitoring		0
17	Spontaneous		
66	Monitoring		0
22	Spontaneous		
38	Monitoring		2
48	Monitoring		4
30	Spontaneous		
22	Monitoring		2
17	Monitoring		0
30	Monitoring		0
28	Spontaneous		
48	Monitoring		0
38	Monitoring		3
30	Monitoring		0
22	Monitoring		0
17	Monitoring		0
28	Monitoring		1
17	Monitoring		1
28	Monitoring		0
48	Monitoring		0
30	Monitoring		0
14	Spontaneous		
38	Monitoring		1
28	Monitoring		1
17	Monitoring		0

14	Monitoring		0
22	Monitoring		2
48	Monitoring		3
58	Monitoring		1
30	Monitoring		4
28	Monitoring		4
10	Spontaneous		
66	Monitoring		0
21	Spontaneous		
37	Monitoring		2
14	Monitoring		1
64	Spontaneous		
38	Monitoring		1
28	Monitoring		1
24	Spontaneous		
2	Spontaneous		
48	Monitoring		1
17	Monitoring		2
53	Spontaneous		
55	Spontaneous		
22	Monitoring		1
14	Monitoring		2
58	Monitoring		1
30	Monitoring		2
28	Monitoring		1
10	Monitoring		0
15	Spontaneous		
66	Monitoring		1
31	Spontaneous		
21	Monitoring		1
37	Monitoring		1
52	Spontaneous		
64	Monitoring		1
38	Monitoring		1
48	Monitoring		2
14	Monitoring		3
24	Monitoring		1
38	Request		

37	Request		
30	Request		
28	Request		
21	Request		
53	Request		
72	Request		
46	Request		
24	Monitoring	21	-12
24	Monitoring	21	-21
24	Monitoring	21	-20
24	Monitoring	21	-20
24	Monitoring	21	-19
24	Monitoring	21	-19
24	Monitoring	21	-18
2	Monitoring		21
24	Monitoring	21	-17
24	Monitoring	21	-16
24	Monitoring	21	-15
24	Monitoring	21	-15
24	Monitoring	21	-14
24	Monitoring	21	-14
24	Monitoring	21	-13
24	Monitoring	21	-13
24	Monitoring	21	-13
24	Monitoring	21	-12
24	Monitoring	21	-12
24	Monitoring	21	-11
24	Monitoring	21	-10
24	Monitoring	21	-9
24	Monitoring	21	-8
24	Monitoring	4	-7
24	Monitoring	4	-6
24	Monitoring	3	-5
24	Monitoring	2	-4
24	Monitoring	2	-3
58	Monitoring		33
66	Monitoring		30
48	Monitoring		27

17	Monitoring		25
38	Monitoring		22
30	Monitoring		20
37	Monitoring		20
14	Request		
2	Periphery		
7	Periphery		
10	Periphery		
22	Periphery		
30	Monitoring		18
30	Monitoring	1	-2
58	Periphery		
30	Monitoring	2	0
30	Monitoring	2	-2
17	Monitoring	8	6
64	Periphery		
30	Monitoring	2	0
30	Monitoring	2	-3
14	Monitoring	6	26
30	Monitoring	2	0
30	Monitoring	2	-3
30	Monitoring	2	-3
30	Monitoring	2	-3
30	Monitoring	2	-3
14	Monitoring	6	3
30	Monitoring	2	-2
66	Periphery		
30	Monitoring	2	-1
30	Monitoring	1	-2
30	Monitoring	2	-2
17	Monitoring	8	6
30	Monitoring	2	-1
14	Monitoring	6	4
30	Monitoring	2	-2
30	Monitoring	2	-3
30	Monitoring	2	-3
30	Monitoring	2	-3
30	Monitoring	2	-3

30	Monitoring	2	-3
30	Monitoring	2	-3
30	Monitoring	2	-2
17	Monitoring	8	6
30	Monitoring	2	-1
30	Monitoring	2	-3
30	Monitoring	2	-3
14	Monitoring	6	3
30	Monitoring	2	-2
48	Monitoring	12	61
66	Request		
30	Monitoring	2	-1
30	Monitoring	2	-6
30	Monitoring	2	-6
30	Monitoring	2	-7
30	Monitoring	2	-7
30	Monitoring	2	-8
48	Monitoring	12	-16
48	Monitoring	12	-26
48	Monitoring	12	-26
48	Monitoring	12	-26
48	Monitoring	12	-26
64	Request		
30	Request		
30	Periphery		
17	Monitoring	10	14
17	Monitoring	10	-9
17	Monitoring	10	-6
17	Monitoring	10	-6
17	Monitoring	10	-5
17	Monitoring	10	-4
17	Monitoring	10	-4
17	Monitoring	10	-4
28	Request		
21	Request		
28	Periphery		
17	Monitoring	10	2
17	Monitoring	10	-3

17	Monitoring	10	-3
17	Monitoring	10	-3
17	Monitoring	10	-3
17	Monitoring	10	-3
17	Monitoring	10	-4
17	Monitoring	10	-4
17	Monitoring	10	-4
17	Monitoring	10	-6
17	Monitoring	10	-6
17	Monitoring	10	-6
17	Monitoring	10	-7
17	Monitoring	10	-7
17	Monitoring	10	-7
17	Monitoring	10	-7
17	Monitoring	10	-7
17	Monitoring	10	-8
17	Monitoring	10	-10
17	Monitoring	10	-11
17	Monitoring	10	-11
28	Monitoring		49
17	Spontaneous		
17	Monitoring		0
28	Monitoring		0
17	Monitoring		0
28	Monitoring		0
17	Monitoring		0
28	Monitoring		0

Bibliography

[AC76] Michael Argyle and Mark Cook. *Gaze and Mutual Gaze*. Cambridge University Press, 1976.

[AFO03] Okan Arikan, David A. Forsyth, and James F. O'Brien. Motion synthesis from annotations. *ACM Trans. Graph.*, 22(3):402–408, 2003.

[AMC03] Matt Anderson, Eric McDaniel, and Stephen Chenney. Constrained animation of flocks. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 286–297. Eurographics Association, 2003.

[ASK+05] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.

[BA00] Victor Blue and Jeffrey Adler. Cellular automata model of emergent collective bi-directional pedestrian dynamics. In *Artificial Life VII, The Seventh International Conference on the Simulation and Synthesis of Living Systems*, 2000.

[BAZB02] Norman Badler, Jan Allbeck, Liwei Zhao, and Meeran Byun. Representing and parameterizing agent behaviors. In *Computer Animation*, pages 133–143, 2002.

[BBT99] Christophe Bordeux, Ronan Boulic, and Daniel Thalmann. An efficient and flexible perception pipeline for autonomous agents. In *Computer Graphics Forum (Eurographics '99)*, volume 18(3), pages 23–30, 1999.

[BC93] A. Bruderlin and T. Calvert. Interactive animation of personalized human locomotion. In *Graphics Interface*, pages 17–23, 1993.

[BC99] Srikanth Bandi and Marc Cavazza. Integrating world semantics into path planning heuristics for virtual agents. In *Intelligent Virtual Agents*, 1999.

[BCGWH04] Simon Baron-Cohen, Ofer Golan, Sally Wheelwright, and Jacqueline Hill. A new taxonomy of human emotions. 2004. To appear.

[BG90] V. Bruce and P. R. Green. *Visual Perception: physiology, psychology and ecology.* Lawrence Erlbaum Associates Ltd., Hove, UK, second edition, 1990.

[BG95] Bruce M. Blumberg and Tinsley A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 47–54. ACM Press, 1995.

[BH00] Matthew Brand and Aaron Hertzmann. Style machines. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[BJ03] David C. Brogan and Nicholas L. Johnson. Realistic human walking paths. In *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, pages 94–101, Washington, DC, USA, 2003. IEEE Computer Society.

[BLCD02] Christoph Bregler, Lorie Loeb, Erika Chuang, and Hrishi Deshpande. Turning to the masters: motion capturing cartoons. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 399–407. ACM Press, 2002.

[BT96] P. Becheiraz and D. Thalmann. A model of nonverbal communication and interpersonal relationship between virtual actors. In *Proceedings of Computer Animation '96*, pages 47–54. IEEE Computing Society Press, June 1996.

[CBC01] L. Chen, K. Bechkoum, and G. Clapworthy. Equipping a lifelike animated agent with a mind. *Lecture Notes in Computer Science*, Volume 2190:72–85, January 2001.

[CCD00] A. Colburn, M. Cohen, and S. Drucker. The role of eye gaze in avatar mediated conversational interfaces. Technical report, Microsoft Research, 2000.

[CH99] Shihkai Chung and James K. Hahn. Animation of human walking in virtual environments. In *CA '99: Proceedings of Computer Animation*, page 4, Washington, DC, USA, 1999. IEEE Computer Society.

[CH05] Jinxiang Chai and Jessica K. Hodgins. Performance animation from low-dimensional control signals. *ACM Trans. Graph.*, 24(3):686–696, 2005.

[Che04] Stephen Chenney. Flow tiles. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 233–242, New York, NY, USA, 2004. ACM Press.

[Cho98] Sonu Chopra. *Where To Look? Automating Some Visual Attending Behaviors of Human Characters.* PhD thesis, 1998. http://www.ircs.upenn.edu/download/techreports/1998/98-17.pdf.

[CKB99] Sonu Chopra-Khullar and Norman Badler. Where to look? Automating visual attending behaviors of virtual human characters. In *Autonomous Agents*, 1999.

[CM81] Peter Collett and Peter Marsh. Patterns of public behavior: Collision avoidance on a pedestrian crossing. *Non-verbal Communication; interaction and gesture, Approaches to Semiotics 41*, pages 199–217, 1981.

[Cou02] M. Coulson. Expressing emotion through body movement: a component based approach. *AISB workshop on Animating Expressive Characters for Social Interactions (Imperial College London)*, April 2002.

[DH03] W. Daamen and S. P. Hoogendorn. Experimental research on pedestrian walking behavior, 2003. 82nd Annual Meeting at the Transportation Research Board, proceedings on CD-Rom.

[Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. In *Numerische Mathematik*, volume 1, pages 269–271, 1959.

[dLL99] Sorabain Wolfheart de Lioncourt and Michael Luck. Motivating intelligent agents for virtual environments. In *Proceedings of the Intelligent Virtual Agents Workshop*, 1999.

[Ekm99] Paul Ekman. *Basic Emotions*. Wiley and Sons Ltd., 1999.

[EMT05] Arjan Egges and Nadia Magnenat-Thalmann. Emotional communicative body animation for multiple characters. In *First International Workshop on Crowd Simulation (V-CROWDS '05)*, 2005.

[EY97] Howard Egeth and Steven Yantis. Visual attention: Control, representation, and time course. In *Annual Review of Psychology*, volume 48, pages 269–297, 1997.

[Feu00] Franck Feurtey. Simulating the collision avoidance behavior of pedestrians. Master's thesis, Department of Electronic Engineering, University of Tokyo, 2000. http://www.logos.t.u-tokyo.ac.jp/~franck/files/thesis.pdf.

[FM98] M. Franz and H. Mallot. Biomimetic robot navigation. Technical Report 65, MaxPlanck Institut fur biologische Kybernetik, 1998.

[For03] Joe Fordham. The Lord of the Rings: The Two Towers: Middle-earth strikes back. *Cinefex*, 92, 2003.

[FP03] Anthony C. Fang and Nancy S. Pollard. Efficient synthesis of physically valid human motion. *ACM Trans. Graph.*, 22(3):417–426, 2003.

[FR84] B. Fehr and J. A. Russell. Concept of emotion viewed from a prototype perspective. *Journal of Experimental Psychology: General*, 113:464–486, 1984.

[FvdPT01] Petros Faloutsos, Michiel van de Panne, and Demetri Terzopoulos. Composable controllers for physics-based character animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 251–260. ACM Press, 2001.

[GB04] Marco Gillies and Daniel Ballin. Integrating autonomous behavior and user control for believable agents. In *Proceedings of the Third international joint conference on Autonomous Agents and Multi-Agent Systems*, 2004.

[GC98] Stephen Grand and Dave Cliff. Creatures: Entertainment software agents with artificial life. In *Autonomous Agents and Multi-Agent Systems*, volume 1(1), pages 39–57, 1998.

[GD05] M. F. P. Gillies and N. A. Dodgson. Attention based obstacle avoidance for animated characters. In *Virtual Reality*, 2005. ISSN 1359-4338.

[Gil01] Marco Gillies. Practical behavioural animation based on vision and attention. Technical Report TR522, University of Cambridge Computer Laboratory, September 2001.

[Gle98] Michael Gleicher. Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, New York, NY, USA, 1998. ACM Press.

[GMHP04] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popovic. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, 2004.

[Gof72] Erving Goffman. *Relations in Public*. Pelican, 1972.

[Gri96] J. Grimes. On the failure to detect changes in scenes across saccades. *Perception (Vancouver Studies in Cognitive Science)*, 2:171–175, 1996.

[GSBS01] M. Garau, M. Slater, S. Bee, and M. A. Sasse. The impact of eye gaze on communication using humanoid avatars. In *ACM SIGCHI*, pages 309–316, 2001.

[GW92] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[Hil84] Michael R. Hill. *Walking, crossing streets, and choosing pedestrian routes: a survey of recent insights from the social/behavioral sciences*. University of Nebraska, 1984.

[Hil99] Randall W. Hill Jr. Modeling perceptual attention in virtual humans. In *Proceedings of the 8th Conference on Computer Generated Forces and Behavioral Representation*, May 1999.

[HKG02] Randall W. Hill Jr., Youngjun Kim, and Jonathan Gratch. Anticipating where to look: Predicting the movements of mobile agents in complex terrain. In *First International Joint Conference on Autonomous Agents and Multiagent Systems*, July 2002.

[HM95] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical Review E51*, pages 4282–4286, 1995.

[HMP05] Bjorn Hartmann, Maurizio Mancini, and Catherine Pelachaud. Implementing expressive gesture synthesis for embodied conversational agents, 2005. Gesture Workshop, Vannes.

[HNR68] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, (2):100–107, 1968.

[HWBO95] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 71–78. ACM Press, 1995.

[IC87] P. M. Isaacs and M. F. Cohen. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In *Proceedings of ACM SIGGRAPH*, pages 215–224. Addison-Wesley, July 1987.

[IK99] Laurent Itti and Christof Koch. A comparison of feature combination strategies for saliency-based visual attention systems. In *Proceedings of SPIE: Conference on Human Vision and Electronic Imaging IV*, volume 3644, pages 473–482, 1999.

[IK00] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506, 2000.

[IKN98] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[Jac95] R. J. K. Jacob. Eye tracking in advanced interface design. In W. Barfield and T. Furness, editors, *Advanced Interface Design and Virtual Environments*, pages 258–288. Oxford University Press, 1995.

[Kah73] Daniel Kahneman. *Attention and effort.* Prentice Hall, 1973.

[KB96] H. Ko and N. I. Badler. Animating human locomotion in real-time using inverse dynamics. In *IEEE Computer Graphics and Applications*, volume 16(2), pages 50–59, March 1996.

[KPN96] R. Knoblauch, M. Pietrucha, and M. Nitzburg. Field studies of pedestrian walking speed and start-up time. *Transportation Research Record (1538)*, 1996.

[KU85] C. Koch and S. Ullman. Shift in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985.

[Kuf98] J.J. Kuffner. Goal-directed navigation for animated characters using real-time path planning and control. In *Proc. CAPTECH '98 : Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, 1998.

[LBB02] Sooha Park Lee, Jeremy B. Badler, and Norman I. Badler. Eyes alive. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 637–644, New York, NY, USA, 2002. ACM Press.

[LCF05] Yu-Chi Lai, Stephen Chenney, and ShaoHua Fan. Group motion graphs. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 281–290, New York, NY, USA, 2005. ACM Press.

[LD04] Fabrice Lamarche and Stephane Donikian. Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. In *Computer Graphics Forum*, volume 23, pages 509–518, September 2004.

[LHP05] C. Karen Liu, Aaron Hertzmann, and Zoran Popovic. Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph.*, 24(3):1071–1081, 2005.

[LP02] C. Karen Liu and Zoran Popovic. Synthesis of complex dynamic character motion from simple animations. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 408–416. ACM Press, 2002.

[LvdPF00] Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. Interactive control for physically-based animation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 201–208, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[LW93] Helmut Lorek and Matthew White. Parallel bird flocking simulation. In *Parallel Processing for Graphics and Scientific Visualization*, 1993.

[Mas03] Massive Ltd. www.massivesoftware.com, 2003.

[MCT01] Jean-Sebastien Monzani, Angel Caicedo, and Daniel Thalmann. Integrating behavioural animation techniques. *Computer Graphics Forum*, 20(3), 2001.

[Met01] Dimitris N. Metaxas. Automating gait generation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 261–270. ACM Press, 2001.

[MT97] Soraia Russe Musse and Daniel Thalmann. A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Computer Animation and Simulations, Proc. Eurographics workshop*, 1997.

[NF04] Michael Neff and Eugene Fiume. Methods for exploring expressive stance. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–58, New York, NY, USA, 2004. ACM Press.

[NI97] S. Nishimura and T. Ikegami. Emergence of collective strategies in prey-predator game model. In *Artificial Life 3*, 1997.

[NRTMT95] H. Noser, O. Renault, D. Thalmann, and N. Magnenat-Thalmann. Navigation for digital actors based on synthetic vision, memory and learning. *Computers and Graphics*, 19(1):7–19, November 1995.

[Old68] S. J. Older. The speed, density and flow of pedestrians on footway in shopping streets. *Traffic Engineering and Control*, 10(4):160–163, 1968.

[OM98] Wilfried Osberger and Anthony J. Maeder. Automatic identification of perceptually important regions in an image. In *International Conference on Pattern Recognition*, August 1998.

[ONPS05] Magalie Ochs, Radowlaw Niewiadomski, Catherine Pelachaud, and David Sadek. Intelligent expressions of emotions. In *1st International Conference on Affective Computing and Intelligent Interaction*, 2005.

[Par98] Raja Parasuraman. *The attentive brain.* MIT Press, 1998.

[Pel05] Catherine Pelachaud. Multimodal expressive embodied conversational agents, 2005. ACM Multimedia, Brave New Topics session, Singapore.

[PLT05] Julien Pettre, Jean-Paul Laumond, and Daniel Thalmann. A navigation graph for real-time crowd animation on multilayered and uneven terrain. In *First International Workshop on Crowd Simulation (V-CROWDS '05)*, 2005.

[PO02] Christopher Peters and Carol O'Sullivan. Synthetic vision and memory for autonomous virtual humans. *Computer Graphics Forum*, 21(4):743–753, 2002.

[PO03] Christopher Peters and Carol O'Sullivan. Bottom-up visual attention for virtual human animation. In *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, pages 111–117, Washington, DC, USA, 2003. IEEE Computer Society.

[Pol00] Tony Polichroniadis. *High Level Control of Virtual Actors.* PhD thesis, University of Cambridge Computer Laboratory, 2000.

[POSB05] Nuria Pelechano, Kevin O'Brien, Barry Silverman, and Norman Badler. Crowd simulation incorporating agent psychological models, roles and communication. In *First International Workshop on Crowd Simulation (V-CROWDS '05)*, 2005.

[PPB+05] Christopher Peters, Catherine Pelachaud, Elisabetta Bevacqua, Maurizio Mancini, and Isabella Poggi. A model of attention and interest using gaze behavior. *Lecture Notes in Computer Science*, pages 229–240, 2005.

[PT05] J. Pettre and D. Thalmann. Path planning for crowds: From shared goals to individual behaviors. In *Eurographics Short Papers*, 2005.

[PW99] Zoran Popovic and Andrew Witkin. Physically based motion transformation. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 11–20. ACM Press/Addison-Wesley Publishing Co., 1999.

[RB05] Ori P. Ratner and David C. Brogan. Simulating crowds with balance dynamics, 2005. Poster at ACM SIGGRAPH 2005.

[Rey87] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proc. SIGGRAPH*, pages 25–34. ACM Press, 1987.

[Rey99] Craig W. Reynolds. Steering behaviors for autonomous characters. In *Game Developers Conference*, 1999.

[Rey00] Craig Reynolds. Interaction with groups of autonomous characters. In *Game Developers Conference*, pages 449–460, 2000.

[Ros08] Edward Ross. *Social Control: A survey of the foundations of order*. Macmillan, New York, 1908.

[SC99] Daniel Simons and Christopher Chabris. Gorillas in our midst: sustained inattentional blindness for dynamic events. *Perception (Vancouver Studies in Cognitive Science)*, 28:1059–1074, 1999.

[Sen65] J. W. Senders. The human operator as a monitor and controller of multidegree of freedom systems. *IEEE Transactions on Human Factors in Electronics*, HFE-5:2–5, 1965.

[SGC04] Mankyu Sung, Michael Gleicher, and Stephen Chenney. Scalable behaviors for crowd simulation. *Computer Graphics Forum*, 23(3), 2004.

[SHP04] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph.*, 23(3):514–521, 2004.

[Sim94] Karl Sims. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM Press, 1994.

[SPS99] Mel Slater, David-Paul Pertaub, and Anthony Steed. Public speaking in virtual reality: Facing an audience of avatars. *IEEE Comput. Graph. Appl.*, 19(2):6–9, 1999.

[SSKO01] Phillip Shaver, Judith Schwartz, Donald Kirson, and Cary O'Connor. Emotion knowledge: Further exploration of a prototype approach. In W. Gerrod Parrott, editor, *Emotions in Social Psychology*, pages 26–56. 2001.

[SSV+99] Danielle R. D. Silva, Clauirton A. Siebra, Jeferson L. F. Valadares, Alessandro L. Almeida, Alejamdro C. Frery, and Geber L. Rarnalho. Personality-centered agents for virtual computer games. In *Proceedings of the Intelligent Virtual Agents Workshop*, pages 99–111, 1999.

[ST05] Wei Shao and Demetri Terzopoulos. Autonomous pedestrians. *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 19–28, 2005.

[Sti00] G. Keith Still. *Crowd Dynamics*. PhD thesis, Mathematics Department, Warwick University, 2000. http://www.crowddynamics.com/Thesis/Contents.htm.

[TCP06] Adam Treuille, Seth Cooper, and Zoran Popovic. Continuum crowds. *ACM Trans. Graph.*, 25(3), 2006.

[Tha01] Daniel Thalmann. The foundations to build a virtual human society. *Lecture Notes in Computer Science*, Volume 2190:1–14, January 2001.

[The93] J. Theeuwes. Visual selective attention: A theoretical analysis. *Acta Psychologica 83*, pages 93–154, 1993.

[TM95] P.A Thompson and E.W. Marchant. A computer model for the evacuation of large building populations. In *Fire Safety Journal 24*, pages 131–148, 1995.

[TMK00] Daniel Thalmann, Soraia Raupp Musse, and Macelo Kallmann. From individual human agents to crowds. In *Informatik/Informatique - Revue des Organizations Suisses d'Informatique*, 2000.

[TT94] Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: physics, locomotion, perception, behavior. In *Proc. SIGGRAPH*, pages 43–50. ACM Press, 1994.

[TWBM97] O. Trullier, S. Wiener, A. Berthoz, and J. Meyer. Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544, 1997.

[UAT95] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *Proceedings of ACN SIGGRAPH*, pages 91–96. Addison-Wesley, August 1995.

[UCT04] Branislav Ulicny, Pablo Ciechomski, and Daniel Thalmann. Crowdbrush: Interactive authoring of real-time crowd scenes. *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 243–252, 2004.

[VP01] Spyros Vosinakis and Themis Panayiotopoulos. Simhuman: A platform for real-time virtual agents with planning capabilities. *Lecture Notes in Computer Science*, Volume 2190:210–223, January 2001.

[Wil66] G. Williams. The effect of target specification on objects fixated during visual search. *Perception and Psychophysics*, 1:315–318, 1966.

[WK88] A. Witkin and M. Kass. Spacetime constraints. In *Proceedings of ACM SIGGRAPH*, pages 159–168. Addison-Wesley, August 1988.

[Wol73] Michael Wolff. Notes on the behaviour of pedestrians. In A. Birenbaum and E. Sagar, editors, *People in Places: The Sociology of the Familiar*, pages 35–48. New York: Praeger, 1973.

[WP95] A. Witkin and Z. Popovic. Motion warping. In *Proceedings of ACM SIGGRAPH*, pages 105–108. Addison-Wesley, August 1995.

[WvG97] Jane Wilhelms and Allen van Gelder. Anatomically based modeling. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 173–180, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[Yar67] A. F. Yarbus. *Eye Movements and Vision.* Plenum Press, New York, 1967.

[YPG01] Hector Yee, Sumanta Pattanaik, and Donald P. Greenberg. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Trans. Graph.*, 20(1):39–65, 2001.

[Zel82] D. Zeltzer. Motor control techniques for figure animation. In *IEEE Computer Graphics and Applications*, volume 2(9), pages 53–59, November 1982.

[ZMCF05] Victor Brian Zordan, Anna Majkowska, Bill Chiu, and Matthew Fast. Dynamic response for motion capture animation. *ACM Trans. Graph.*, 24(3):697–701, 2005.