

Eye movements and attention for behavioural animation

By M. F. P. Gillies* and N. A. Dodgson

*Correspondence to: M. F. P. Gillies, UCL at Adastral Park,
Ross Building pp1, Adastral Park, Ipswich IP5 3RE, UK.
E-mail: m.gillies@cs.ucl.ac.uk

This paper describes a simulation of attention behaviour aimed at computer-animated characters. Attention is the focusing of a person's perception on a particular object. This is useful for computer animation as it determines which objects the character is aware of: information that can be used in the simulation of the character's behaviour in order to automatically animate the character. The simulation of attention also determines where the character is looking and so is used to produce gaze behaviour.

KEY WORDS: computer animation; autonomous characters

Introduction

Gaze patterns are one of the most expressive aspects of human outward behaviour, giving clues to personality, emotion and inner thoughts. Simulating the eye and head movements of a person as they look around their environment is vital to creating a convincing character. A character can move and act highly realistically, but if their gaze is fixed and unmoving they will seem lifeless and inhuman. Film makers make great use of their character's patterns of gaze to suggest their thoughts, and gaze can be vital in how we judge other people. It is thus important that there are a variety of attention behaviour patterns possible for different characters. There has been some work on simulating people's gaze in conversational and social situations, as described in the next section; however, there has been little on simulating gaze in other situations. Garau et al[1] have studied how people are affected by avatars' eye movements. It was found that appropriate eye movements increased the user's engagement with the avatar but random eye movement was not helpful. The experiment was with face-to-face conversation and so not necessarily relevant to all applications, but it does show the importance of a good attention model.

Attention is integral to simulating gaze. It consists of the focusing of a person's perceptual and cognitive capacities on a particular object or location. Though we are generally aware of the environment around us, we only attend to or 'look at' one place at a time. Our perceptions are more detailed at the focus of our attention and we are more likely to be aware of and remember events that occur at the focus than away from it (see Pashler[2] for an overview). This means that what we are attending to has a great effect on our perceptions and therefore attention is important for simulating behaviour for animation. In simulating behaviour it is important to take account of which objects the character is aware; otherwise the character will react to objects and events that it does not know about. Vision is important in determining which objects the character is aware of but it is wrong to think that what the character is aware of is the same as what is in its visual field. Though people are always aware of objects at their centre of vision, awareness in

the periphery is variable. Some features, such as motion, 'pop out' and are obvious; others do not. Therefore it is not enough merely to simulate the visual field; it is important to simulate where the character is looking in the visual field. The simulation system we describe does this by having a sequence of foci of attention. The periphery of vision is simulated by certain peripheral objects capturing the character's attention. Once an object has been attended to it is stored in a list of objects that the character is aware of. It is thought that the function of attention in humans is to make efficient use of cognitive resources by applying them to one object at a time. Happily, a simulation of attention can perform a similar role in behavioural animation as many calculations and tests only need to be performed on the focus of attention or on the objects the character is aware of, and not on all the objects in the visual field. In many ways this work is a development of Chopra-Khullar and Badler's,[3] as discussed below, though there are a number of changes and improvements.

The architecture is aimed at both autonomous and semi-autonomous characters. For autonomous characters it is used to provide eye movements and to provide information of what the character is aware of. For semi-autonomous characters it can provide autonomous eye movements which can supplement non-autonomous aspects of behaviour and generate autonomous behaviour which can be used for high-level commands. It is important that behaviour which is generated autonomously at the time of animation can still be controlled by a human animator. It is no use having autonomous behaviour that is unsuitable for a designer. This control should happen prior to the animation being run. For this purpose we provide various parameters that can be adjusted to shape how the attention behaviour is performed. All of the changes that the designer can make to the attention manager are done through these parameters and they can be performed offline before the character is used, thus making them suitable for autonomous characters.

Previous Work

Eye movement during conversation has been simulated in a number of ways as a form of non-verbal communication. Examples include the work of Thórisson[4], Vilhjálmsson and Cassell[5] and Colburn et al.[6]. However, it has not been studied often in other contexts nor as a full attention system that is integrated with behaviour. Hill[7] has a simulation of attention for a virtual helicopter pilot that includes attention capture, selective attention based on features

of objects and their importance to the character's task, and perceptual grouping of objects. However, it is applied to a helicopter in a military situation, not to the animation of a human figure, and it does not produce animations of eye and head movements.

Rickel and Johnson use gaze to indicate attention for their virtual tutor[8]. For example, the tutor looks at an object when manipulating it, or at the user when talking to them. The tutor also looks at the user to indicate that it is evaluating their performance and can use gaze to indicate an object to the user. This model of gaze is strongly tied to the task of tutoring and, to some degree, interpersonal behaviour in general. Many features are more didactic than realistic; for example, it is not realistic to look constantly at an object while manipulating it (unless the manipulation requires great concentration); however, in this application, it helps to focus the user's attention on it.

The only previous work that tackles the same problems as the present work is by Chopra-Khullar and Badler[3]. They developed an architecture for producing gaze behaviour in a computer-animated character. It is similar in scope to the work described here, and our use of a request-based system is based on their work. However, the current work provides a number of improvements. It uses two queues: one representing intentional gaze behaviour (gaze behaviour requested by other behavioural controllers); the other representing peripheral attention capture, which in their system is caused only by moving objects. They also model spontaneous looking, an idling attention behaviour which is analogous to undirected attention behaviour discussed here, though the method is different, as described below. One aspect of Chopra-Khullar and Badler's work is the modelling of the degradation of performance as cognitive load increases. This is not tackled by the current work and would be an interesting extension.

The authors' initial experiments with simulating attention used a queue system similar to Chopra-Khullar and Badler's. However, there are disadvantages to this: it is unsuited to producing timely gaze shifts and not well suited to producing general monitoring behaviour. Certain gaze behaviour is closely connected to a task and has to occur at exactly the right time, synchronized with the task. The problem with a queued system is that it is difficult to make sure a gaze shift occurs at an exact time. If the queue is not empty when the request is issued it will be delayed behind other requests. Even if the queue is empty the eye request may be preempted by a peripheral event and so be delayed. These delays can result in the eye behaviour associated with a task happening after the task is finished, which is clearly undesirable. It is therefore better to handle timely events as special cases. Even for looks that are not time constrained the queue system is not optimal. The problem here is that most such gaze behaviour does not consist of a one-off look but rather an interest in an object or location to which the actor will often return. It is clumsy to produce this sort of behaviour with a queue-based system as behavioural controllers have to add new requests to the queue at arbitrary intervals; they also have to keep track of which requests have been sent in case they stop being relevant and so have to be removed from the queue (a common situation). We believe that it is better to build a system around monitoring which involves just a single monitor request and a single purge request.

Observations of Human Behaviour

Our model of attention is based on a number of features of human attention behaviour, some of which are found in the literature of gaze behaviour (particularly Argyle and Cook⁹ and Yarbus¹⁰ and some of which come from informal observation. Our original model was based mostly on Chopra-Khullar and Badler's, with additional input from the work of Argyle and Cook and of Yarbus. However, we found that there were many aspects of attention behaviour that have not been adequately studied. We have therefore made a series of informal observations of people in natural situations. They were aimed at rectifying problems with our initial model and at finding new types of behaviour that the model did not handle. The main observations are described below:

- People will generally maintain a constant angle of vision to the horizontal and normally only rotate their direction of gaze in the horizontal plane. This angle will generally point slightly downwards or, less often, straight ahead. People rarely look up unless they are looking at something specific.¹
- There is a wide variation in the length of looks but it is notable that the distribution appears to have two peaks. People will tend to intersperse very short looks (which we will henceforth call glances) with longer looks (which will be called gazes).
- People, when walking down the street, will often have a behaviour pattern where they will have long forward gazes interspersed with short glances elsewhere in the environment. This is likely to extend to other tasks, where a person will gaze at the focus of the task interspersed with glances at other locations. The opposite behaviour is also common; people spend most of their time gazing around while occasionally glancing ahead of themselves, presumably to ensure they do not walk into something. People seem to exhibit either one behaviour or the other.
- In general, people tend to return their gaze repeatedly to the same place, looking a number of times at something that interests them (we will call this monitoring). Yarbus notes this behaviour in people whose eyes

¹This is an important feature of human gaze behaviour. The difference between a virtual human with varying gaze angle and a real person is very noticeable; in fact, adding this feature probably produced the greatest improvement in realism of any feature. However, it is a feature that does not seem to have been mentioned in the literature. Garau noted the problem of varying vertical gaze angle, producing unrealistic behaviour in her experiments, but confirmed that it had not been studied (private communication and Garau et al.[1]).

have been tracked while looking at pictures (Yarbus [10] p. 194).

- Though the previous discussion has been in terms of eye movements, the direction in which someone's eyes are looking can be hard to see. People's attention is normally seen through the orientation of their head. Apart from small changes of direction people will generally change the direction in which they are looking by moving their head. The actual eye movements only become important when the character is in close-up or

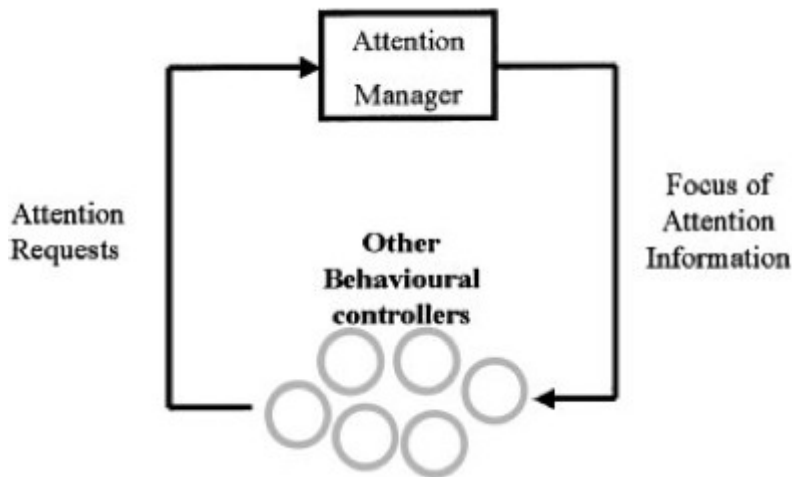


Figure 1: The attention manager and other behavioural controllers. The attention manager receives requests for attention shifts from other behavioural controllers. It arbitrates between them and eventually executes them. They change the focus of attention, which can then be used by the other behavioural controllers.

where the character is looking in the general direction of the viewer; in which case people are very good at determining whether or not someone is looking directly at them.

Attention Architecture

This section describes the attention mechanisms themselves. They are based on a set of collaborative agents generating behaviour, known as behavioural controllers, that collaborate to produce the final behaviour of the character. The behavioural controllers are shown in Figure 1. The attention manager is the main behavioural controller of the attention architecture. It controls and arbitrates the attention behaviour and controls the eye movements of the character. Other controllers send requests for attention shifts to the attention manager.

In addition to this mechanism, the user can request that characters make attention shifts. By clicking on an object the user can bring up a menu that includes various options for looking at the object, for example, glancing at it or monitoring it.

As well as receiving requests from other behavioural controllers the attention manager sends information about what the character is looking at to other controllers. These controllers can react to this information while producing behaviour. The focus of attention is passed to other controllers for use in their behavioural algorithms.

The Attention Manager

The attention manager is the main behavioural controller involved with eye movement and attention. It performs various functions. Its major function is to supply a series of gaze directions to the low-level eye movement controller and a series of foci of attention for those behavioural controllers which rely on the attention of the character. In order to do this it must manage the various requests for eye movements and attention shifts made by other controllers and arbitrate between them, choosing a single one at any given time. It also has a secondary function of generating undirected attention shifts, which will be discussed later.

Like Chopra-Khullar's system, the attention manager receives requests which are then processed to produce attention shifts and eye movements. However, the requests are processed in a different way to Chopra-Khullar's. The next section describes the structure of the attention requests and the section 'Processing the Requests' describes how they are used.

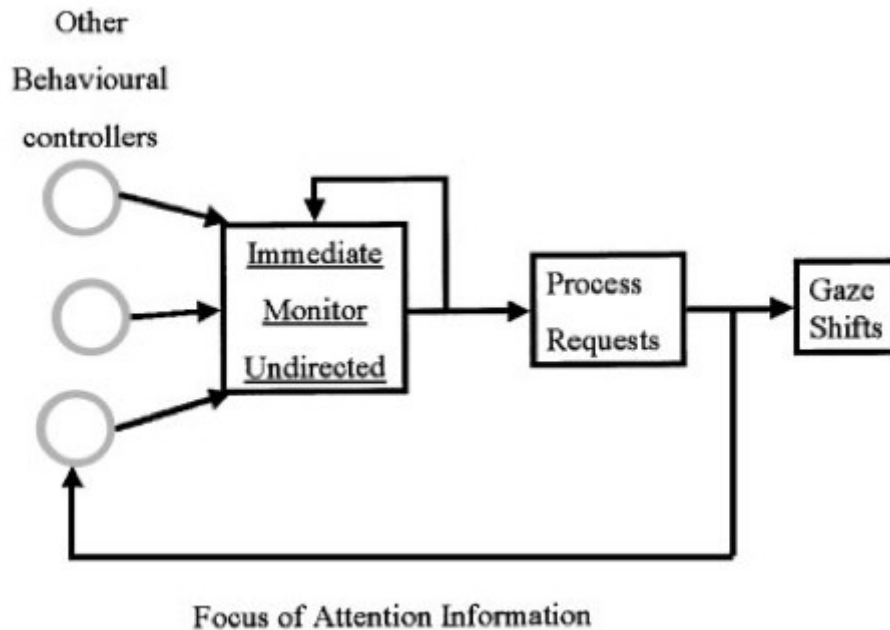


Figure 2: The attention manager receives requests of two different types: immediate and monitor requests. It chooses between these, or if none are present performs undirected attention to generate a request. When a request has been chosen it must be processed to turn it into an attention shift and eye movement. If necessary it is then sent to the gaze shift behavioural controller, which moves the character's eyes.

Figure 2 gives an overview of the attention manager. Requests can be sent to it at any time by other behavioural controllers. When the attention manager is ready it processes one of these requests. It does this by arbitrating various requests as described in the section 'Types of Attention Behaviour'. It then transforms the request into actual attention behaviour, possibly moving the eyes as described in the section 'Processing the Requests'. This attention behaviour will last for a length of time. When it is finished the attention manager will choose and process another request. This timing can be overridden by another controller which can request that an attention shift occurs immediately, as described in the section 'Immediate Shifts', in which case the current attention behaviour is interrupted.

The attention manager has various parameters that can be changed by a character designer in order to alter the character's behaviour, for example, the mean and variance of gaze length or the probabilities of performing various actions. The parameters themselves are described in the appropriate sections.

Attention Requests

The attention manager receives requests for shifts of attention from other behavioural controllers. Each request has a pointer to the controller that made the request, so that the attention manager can notify the controller when the character is attending to the request or if the attention manager fails to enact the request (the section 'Processing the Requests' describes which requests are rejected). The requests can be of three types, depending on what the character should attend to:

- *Location requests* represent a location in the world. They are specified by a position vector in world coordinates. As the character moves, it moves its eyes to compensate for its own motion and so keeps fixating the same absolute position. This is used in situations where the character has to look at a specific place in the world but one that does not easily correspond to an actual object, for example, the character's destination while walking.
- *Local requests* are the simplest type of request; they consist of a vector that represents a direction in local coordinates relative to the character's head. Thus they are defined relative to the character as opposed to globally in world coordinates. As the character moves, its eyes do not move but keep looking in the same direction relative to the character's head. This type of request is useful for situations where the character is looking at the environment in general, rather than at specific places or objects. An example of their use might be scanning the street ahead for obstacles or looking around, as in undirected looking (see below).
- *Object requests* are requests to look at a specific object in the environment and are specified by the identifier of

the object. The advantage of this form of request is that the eye manager has access to the object itself. This access is via an interface that makes available two types of properties: standard geometric and higher-level properties. The former are the sort of standard geometric properties that would be perceivable by the character, such as position and velocity. The latter can be added by the animator and are represented as tags; these might include the property of being interesting or the property of looking like a cup. The attention manager can use the object's properties for various activities, for example, to track its moving position. It can also pass the object on to other behavioural controllers that can use it to perform various visual algorithms. Object requests are the type that is most used by specific tasks as they generally require attention to a given object, for example, if the object is being stepped over.

Parameters of Attention Requests

As well as containing the location or object to be attended to, a request also specifies various aspects of the character's attention behaviour while attending to that object. These are specified in terms of flags and pieces of extra data to determine their effect. These include:

- *Glance*, which is a tri-value flag determining if the eye motion should be a short glance, a long gaze, or don't care, in which case the attention manager decides.
- *Interval*, which applies to requests that represent the character occasionally monitoring a location. It is an approximate time between looks.
- *Minimum distance*, which is the closest an object can be for the character to still look at it (see the section 'Reject Invalid Requests' for details).
- *Maximum times*, which is the maximum number of times a character will look at the target of a monitor request. This is normally infinite.

Types of Attention Behaviour

There are two types of attention behaviour that can be requested by behavioural controllers, as illustrated in Figure 2. Immediate attention shifts make the character move its attention to the target as soon as the request is received.

Monitoring behaviour makes the character look at the target occasionally. Finally if there are no requests from other controllers the attention manager generates undirected attention behaviour, analogous to Chopra-Khullar's spontaneous looking. This is a form of idling attention shifts. The attention manager arbitrates between the types of behaviour using strict priorities. Immediate requests have the highest priority, above monitoring. Undirected looking has the lowest priority, happening only if there is nothing else for the character to attend to.

Immediate Shifts

Immediate shifts are the simplest form of attention behaviour. If the attention manager receives a request for an immediate shift it produces an attention shift in the same frame for animation, delaying any other pending requests for attention behaviour. To prevent an immediate request placed just after another one from overwriting the first, the immediate request channel is locked when a controller places a request on it. This means that, if a controller attempts to place an immediate request while another is active, it is unable to do so and is notified. The controller can then wait for the channel to become free and if necessary delay any behaviour that has to be synchronized with the attention shift. The channel remains locked until the character is actually looking at the object, i.e. it has finished moving its head and eyes towards the target. This is an improvement on using a queue to schedule multiple requests as Chopra-Khullar and Badler do, as it deals with simultaneous multiple requests while still allowing attention behaviour to be synchronized with other behaviour. If the character's attention is caught by a sudden noise while about to perform a task that requires its attention it will wait until it can attend to the task before performing it. It will not perform the task and then attend to it at a later time.

Monitoring

It is often the case that a character is interested in a particular object or location and needs to attend to it but not necessarily at a particular time. It is also the case that if a character is interested in an object or location it will be interested in it for a period of time and will want to attend to it more than once. Monitoring is this sort of behaviour. Monitoring requests are sent to the attention manager by behavioural controllers and are placed by the attention manager in an array. When the character has finished attending to a location or object this array is tested to see if the character should attend to one of the monitoring requests. The requests are chosen with a frequency determined by the interval parameter. If the time since a request was last chosen is greater than interval it is chosen again. When more than one request has been inactive for longer than its interval the request that is the longest time past its interval is chosen.

A monitoring request can be removed for a number of reasons. There is a parameter in the request that specifies the maximum number of times the character should look at the target. When this has been exceeded the request is deleted. Normally this is set to infinity so that the character monitors the target until told to stop. However, it can be set to a small number so that the monitoring mechanism can be used to implement the case where the character needs to look at a target only once or twice but without exact time constraints (i.e. not an immediate request). Another parameter in the request is the minimum distance; this makes the character stop monitoring an object that is less than a certain distance in front of it. Some objects are important enough that the character should turn around to monitor them but many are not and should be rejected when they pass behind the character. This is common in a large number of circumstances; for example, when walking down a street (or driving) it is normal to ignore things once they have been passed. Also in

social situations, such as parties, it is acceptable to look at a person one is not talking to occasionally but not to turn around or make obvious bodily movements to look at them. Finally, if a person is monitoring something only out of a vague interest, they might not want to turn around to look at it. The parameter is normally set to a small distance in front of the character so that the character no longer attends to targets that have passed behind it or are about to. It can also be set so as to include targets some way behind the character or only those that are far from the character (this last can be used for monitoring people on the street, where it is socially acceptable to look at people from a distance but not close up). Finally, behavioural controllers can request that a particular item can be removed from the monitoring array. This can be when a controller is making the character monitor an object for some reason but then finishes that task and so is no longer interested in it. It can thus be seen that this monitoring mechanism allows for a wide range of behaviours to be simulated.

Undirected Attention

Undirected attention is the pattern of looking and attending that is produced when the character has no definite attention required by its behaviour, i.e. no request sent by other behavioural controllers. It is how the attention manager chooses what to do when there is no immediate request and all the requests in the monitor list are dealt with. It is analogous with the term spontaneous looking used by Chopra-Khullar and Badler[3] based on a term of Kahneman[11]. Our method for choosing where to attend to is, however, different from Chopra-Khullar and Badler's.

Chopra-Khullar and Badler's approach is image based; the scene is rendered to an image from the point of view of the actor and this image is used to determine places of interest. Areas of the image where the difference between the colours of neighbouring pixels is large are considered interesting and so the actor will look in that direction. Although this can produce suitable looking patterns this is not always a good heuristic. For example, an actor might be walking in a park over grass which is highly textured and so is likely to have a large pixel difference. During the walk the actor might pass a minimalist sculpture that is very smooth and so have a low pixel difference. However, the character's attention is more likely to be drawn to the sculpture than the ground. In fact, Yarbus states that, based on his experiments on tracking the eyes of people looking at pictures, there is no connection between features of the image of an object and whether someone will look at it. Complexity is not a factor:

any record of eye movements shows that, per se, the number of details contained in an element of the picture does not determine the degree of attention attracted to this element. This is easily understandable, for in any picture, the observer can obtain essential and useful information by glancing at some details, while others tell him nothing new or useful. (Yarbus,[10] p. 182)

He reaches the same conclusion about colour, brightness and contours. The only factor he accepts is the amount of information the observer can extract from a feature.

In general it is not feasible to find a heuristic to determine what the character finds interesting to look at, as the reasons for finding something interesting are so varied. Instead we have tried to use a more general approach which allows the animator more control over undirected attention patterns.

There have in fact been two general types of behaviour observed in people moving or standing still in an environment without having a very definite object to attend to (see above). Some people tend to look forward and often slightly downward without moving their gaze around much. Others are much more likely to look around themselves at their environment. The looking-forward behaviour is probably dependent on the fact that most of the people observed were on a street and mostly walking around. Different tasks might have a different default direction; for example, a person eating is likely to look at their plate. Thus the forward behaviour can be modelled as looking in a default direction. Using these observations as a basis, our attention manager chooses between these two behaviour patterns with a probability that can be set by the character designer.

If the choice is to look at the default direction a local attention request is generated; this results in the character looking in that direction. There is no random variation in this gaze pattern as this form of behaviour seems to involve fairly constant gaze patterns. The default direction will normally be forwards and slightly downwards, unless some other behavioural controller overrides it.

If the character looks around the environment its attention can be captured by an interesting object. This is different from the attention capture by important objects described in the section 'Peripheral Vision and Attention Capture', where the attention must be captured by an object which is relevant to the current task. That is directed attention. In undirected attention capture there is no particular reason to look at an object other than a vague interest. Each object has an interest value associated with it which determines how probable it is for a character to look at it. For example, an animator might want to put a statue in a crowd scene and give it a high interest value so that many characters in the crowd will look at it. The undirected attention procedure chooses which object to look at by choosing one at random from a set of objects it is aware of and then accepting it with a probability equal to its interest value. This results in every object being chosen with a frequency proportional to its interest value.

If there are few interesting objects in the environment, choosing objects will result in constant repetition of the same gaze directions or even difficulty in finding an object to look at. For this reason the character can also look at random locations around itself. This either happens by the system occasionally choosing at random not to look at an object or when it fails to find an interesting object after 15 attempts. If this happens it generates a random gaze angle in the horizontal plane. This angle is somewhere in the 180 arc in front of the character. A rotation around the left to right axis

is also generated. This can be a random angle but if the character has a tendency to keep a preferred gaze angle to the horizontal it might be this angle. These two rotations are then combined to give a local request.

Processing the Requests

The various attention behaviours will result in a request that must be turned into an actual attention shift and possibly an eye movement. This involves a number of steps as shown in Figure 3.

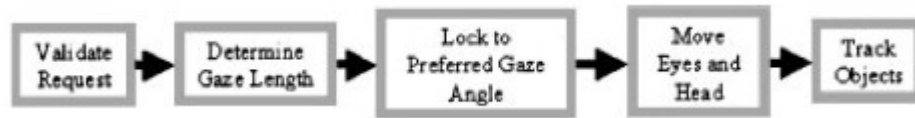


Figure 3: The sequence of actions that are performed on an attention request to execute it. These are the steps that must be taken to create an actual attention shift and eye movement from a request.

Reject Invalid Requests

The first task is to test whether the request is valid. There are three reasons why the request could be invalid. First, if the location is nearer than the request's minimum distance (see the section 'Parameters of Attention Requests' above). This is a simple test which is used to model a number of effects. Often a character will be willing to monitor objects that are in front of it but be unwilling to turn around to look at them. Objects that are very close in front should also be rejected, for example, a cup when drinking. Also it might be socially acceptable to look at a stranger from a distance but not close up. The second reason is that the object is not visible because it is occluded. Finally, a behavioural controller can specify that the character has to keep its head still, for example, if the character is eating and putting food into its mouth. If the character has to keep its head still it cannot look at targets that require a head turn (i.e. if the direction of the request is further than a certain angle from the current gaze direction) and so these targets are rejected. If any of these tests fail, the attention manager must find a new request in the same way. The controller that made the request is notified if it fails. Otherwise the request is successful and the current focus of attention is set to be the location or object of the request.

Length of Gaze

The second step is to determine certain attributes of the request. There are two main mechanisms that control the length of gaze. Firstly two categories of length are defined: short glances and longer gazes. A request can include a flag defining what length it should be. This allows other behavioural controller high-level control over the gaze. For example, if a controller requires that the character look at something surreptitiously it can request a glance (probably with the flag set for keeping the head still). On the other hand, if a controller requires concentration on an object it will request a gaze. The actual length of the gaze is determined at random, with different means for glances and gazes. The means themselves can be set by the character designer, thus varying the lengths of gazes for the character as whole, altering its perceived personality.

When the request does not specify the length of the look, the attention manager must determine it. This is based on the target. We would like to implement a number of different features that produce different lengths (and allowing the character designer to add to them). Currently only one is implemented and this depends on location of the target relative to the character. The probability of glancing is different if the target is near the preferred gaze direction (normally in front of the character) than if it is not. Depending on how the character designer sets these probabilities, the character can be made to look forwards and only glance at its surroundings or vice versa, both behaviours that are common among people walking down the street. Allowing different criteria for determining the probability of glancing would allow for interesting behaviours, especially when dealing with other characters, for example, always glancing at other characters might indicate shyness or embarrassment.

Preferred Gaze Angle

It has been noted (see the section 'Observations of Human Behaviour' above) that people tend to have a preferred vertical gaze angle. This will be the angle to the horizontal of the character's default gaze direction (see above). It can be set by the character designer for different characters. Each character has a probability (another user set parameter) that they will maintain this angle. If they do try to maintain this angle it must be checked that it is possible to maintain it by testing the height of the object being looked at. If it is not possible, the angle will be changed.

Moving the Eyes and Head

Once the details of gaze are determined the attention manager must actually make the character look at the object or location. This is done via a behavioural controller which controls the orientations of the eyes; the eyes are rotated so as

to point at the target. This is very simple but not sufficient. Certainly for some targets, just moving the eyes without moving the head is enough (Figure 4(a)). However, if the target is far from the forward position the necessary rotation might be too large to be convenient without moving the head (Figure 4(b) and (c)). If the rotation is very large, rotating the shoulders is also necessary (Figure 4(d) and (e)). An added impetus for head and shoulder rotation is that, if the character is viewed from far away or not straight on, the eyes can be too small to show where the character is looking. The model used in the diagrams has enlarged eyes relative to a real person but the direction of gaze becomes unclear even at moderate distances. With an accurate model the situation would be worse. However, there are times when a head turn is not desired, for example, if the character is eating or it is being surreptitious; in this case the requesting controller can set a flag that prevents the character moving its head. Also, the threshold at which a character moves its head varies so that characters can be made to move their head less often. Another control that the character designer has is speed of rotation of the head or shoulders.

Thus it is important to turn the head. There are two threshold gaze angles for moving the head; the threshold for horizontal and vertical angles is different. If the gaze angle is within the threshold for the current head position the character will not move its head. Also, the character will tend to return its head to the central, forward-facing position; therefore if the gaze angle is within the threshold for the central position the character's head will move back to the central position. There are different, greater threshold angles for moving the shoulders. These thresholds can vary between characters.

The head is moved by rotating it so as to point its local forward axis towards the target. The shoulders are turned by half that amount, so that they are angled half way between the forward direction and the target. This half shoulder turn produces a more natural result than either no turn or a full turn (see Figure 4(e)).

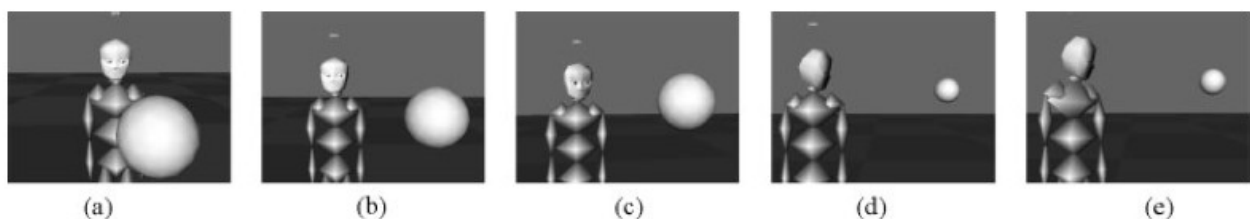


Figure 4: Though moving the eyes is sufficient in some cases (a) looking in some directions can be awkward (b) without turning the head (c). It is sometimes also necessary to rotate the shoulders: (d) shows just the head been turned, while (e) shows both head and shoulders being turned to give a more natural look.

Tracking Objects and Locations

Finally, if either the object being looked at or the character is moving, the character's gaze must follow the object or location. This is done by updating the eye's fixation point every frame. If the character's head is already moving or if the angle of gaze exceeds the threshold while it updates then the head's rotation is also updated.

Peripheral Vision and Attention Capture

While the focus of attention is directed to a single location at any given time, the character needs also to be aware of events in the periphery of vision. This is defined as being anything within 90° of the centre of vision. Though, in general, objects in the periphery are ignored, relevant objects can capture the attention of the character. Unlike Chopra-Khullar and Badler's system, where attention is only captured by moving objects, there are a range of possible relevant or interesting objects that can capture attention. These can vary from task to task. For example, in the task of walking in a cluttered environment, relevant objects are considered to be those which are moving and those which are in the path of the character. These are the objects with which there might be a collision. Attention capture is the main mechanism by which the character becomes aware of objects.

Various behavioural controllers perform the attention capture by peripheral vision depending on what is capturing the attention; for example, controllers of the walking group perform attention capture in the cases described above. They scan objects in peripheral vision and check if they are relevant to the character's current action. Some special-purpose controllers detect specific geometric properties such as moving objects, or objects in the character's path. There are more general peripheral vision controllers which search for user-defined properties. These can represent any property of an object and consist of a tag attached to an object with a name and a value between 0 and 1, for example ('shiny', 0.7). The general peripheral vision agent searches for objects for which a tag with a particular name is defined. This allows controllers to react to objects with a user-defined value. This sort of peripheral vision agent can be added to the character dynamically to produce new behaviour patterns. The applications section below gives an example of how a new peripheral vision agent can be created to make a character react to objects with a certain property. If an object is selected, a shift of attention might be requested from the attention manager. Some objects require a fast reaction, as they

are imminently approaching the character (i.e. their distance from the character divided by their relative speed is low²). These objects are automatically passed to the attention manager as immediate requests. If the object is not imminently approaching, the peripheral behavioural controller will either, with a set probability, send a monitor request or do nothing but add it to the list of objects the character is aware of. This ensures that the character will become aware of it, even if it does not look at the object. If there are two possible targets the one with the most imminent object takes precedence. If an object is not dealt with (looked at and reacted to in a behaviour pattern/object-dependent way) it will become more imminent and the request will become an immediate request. This could happen in a situation where there are a large number of immediate requests which prevent monitoring requests being processed. This should be a rare situation as immediate requests are designed only to be used occasionally. Attention capture, together with the undirected attention, ensures that the character is aware of its surroundings.

Examples and Applications

This model of attention has been applied to different situations and types of behaviour. This section describes examples of it in action. The first examples show just the eye movements produced by the attention model. The other two examples, in their own sections, describe behavioural algorithms that have been built around the attention model. These will be described more fully in further publications.

Figures 5–7 show the character walking between two rows of columns. These examples rely only on undirected attention to generate the behaviour and show the effect of different parameter settings of behaviour. In Figures 5 and 6 the character has been set with a high probability of looking around itself in undirected attention and a high preferred gaze angle. In Figure 7 the settings are the opposite. The parameters are set before the animation starts and then the behaviour is generated autonomously. This method is therefore suitable both for offline animation systems and autonomous characters.

Navigating an Environment

The first application of the attention model is navigation of an environment. Traditionally this has been done either by path planning, normally precomputed and better suited to a static environment, or reactive planning, which deals well with moving objects but tends to have problems with complex environments. Increasingly it is being realized that it is important to combine the two: using planning to choose a rough path around the large, static obstacles while smaller or moving obstacles are avoided reactively as the character becomes aware of them. For this sort of system attention is very important. When and how a character reacts to an obstacle depends not only on the relative positions and velocity of the character and object but also on whether the character is looking in the direction of the object.

Using the attention model makes it possible to build a system where the character's reaction to obstacles seems appropriate to the direction of the character's gaze, and also the character's gaze seems appropriate to the environment and its movements. Figure 8 shows some frames of navigation behaviour. This application has been implemented as an autonomous behavioural system and will be described in more detail in a forthcoming publication; here we give only a brief description of how it uses attention.

Whenever the character attends to an object the object is passed to a behavioural controller that detects whether the character is on a collision course with the object. If so the object is then passed to other controllers that take action to avoid the collision. The navigation agents also send requests to the attention manager to look at the objects that they are dealing with. A peripheral vision controller ensures that objects with which the character might have a collision are detected, for example, moving objects. This illustrates the working of the attention model. Peripheral vision detects objects that are sent to the attention manager so that the character attends to them. Objects that are attended to are then sent to other controllers which react to them. Finally these controllers can send back new requests to look at objects or locations that they are dealing with.



Figure 5: A character walking between two rows of columns, demonstrating undirected attention. The gold column with a sphere on top is classed as more interesting by its object features. The parameters of the character are such that its looks around itself more than looking forwards and has a high gaze angle.

² Lee[12] presents this measure (called Time-To-Contact) as the way in which people judge collisions and interceptive action.

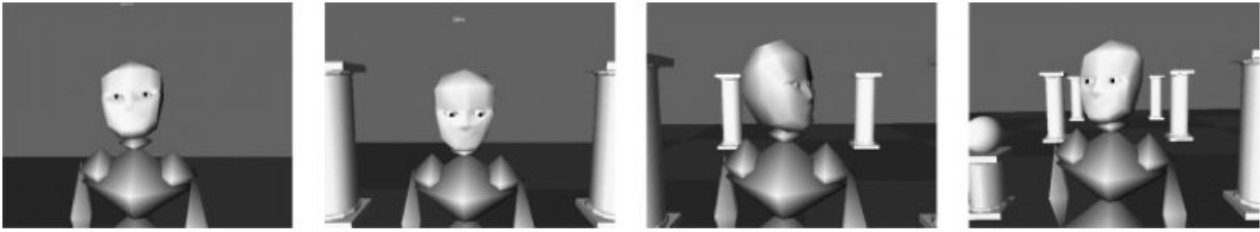


Figure 6: A close-up showing the eye movements from Figure 5.

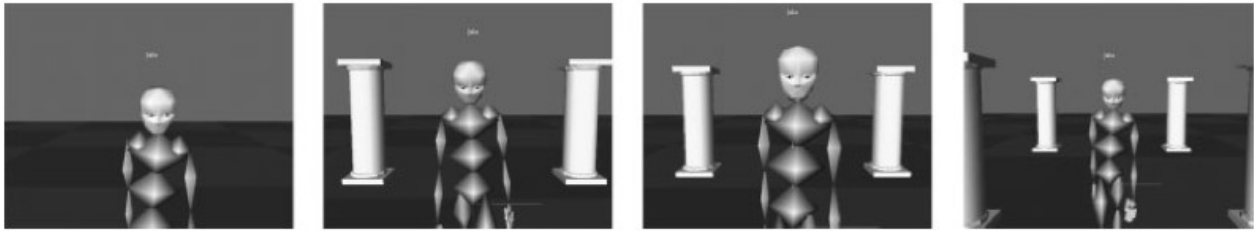


Figure 7: A character walking in the same environment as Figure 5 but with different parameter settings: a large tendency to look forwards with a low gaze angle. The character's gaze only occasionally moves from the ground in front of it. Its gaze raises slightly in frame 3. Frame 1 is interesting: the character looks up without raising its head.

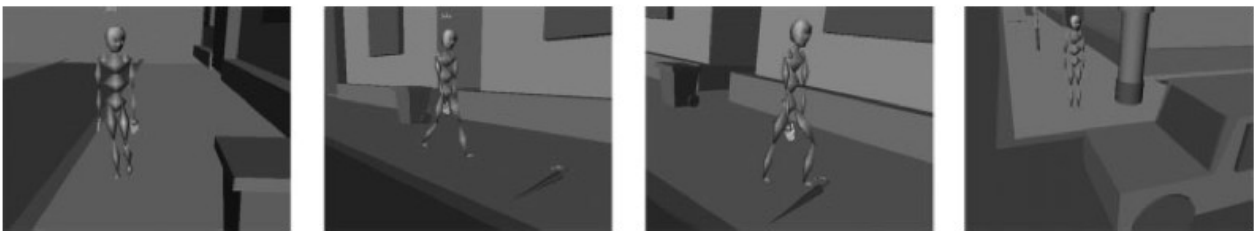


Figure 8: A longer example of a character walking along a street. It changes its path to avoid colliding with the bin, steps over the umbrella and stops to let the car go past. What the character is aware of is determined by the attention mechanism and the character's gaze behaviour is appropriate to the rest of its behaviour.

Simple Actions

The second application is a more general one. In it attention behaviour is added to simple actions. These actions are based on pre-existing motion and consist of small pieces of motion that have some target which is manipulated or acted on. Examples are drinking from a cup or catching a ball. These actions are designed to be the sort of action that would be requested by the user for a user-controlled avatar or which would be the building blocks of behaviour for an autonomous character.

The attention simulation is added to the actions in order to produce appropriate gaze behaviour. The actions themselves can be designed by non-programmers from pre-existing pieces of motion. As the designer will have their own idea of the nature of the action, it is desirable to give a good deal of control over the gaze behaviour at design time while hiding this control when the action is invoked (by a user or a higher-level behavioural routine) in order to reduce complexity of control. The designer of the action adds gaze behaviour by tagging the action with attention requests. When the action is created it is divided into a number of periods representing important moments in the action. A number of targets are also added; these might be objects that the character is actually manipulating or touching, like the cup when drinking, or other objects, for example, while drinking, the character might be in conversation with another character who would be a target. The designer can tag the beginnings of periods with attention requests so as to make the character start or stop monitoring something, or to look at it immediately. The request will refer to one of the targets of the action. For example, in Figure 9 the action is tagged with an immediate request to look at the ball being picked up at the beginning of the period where the character picks it up. The designer can control the parameters of the request setting, for example, whether it is a glance or a gaze or whether the head should keep still. This allows the designer to create a range of different gaze behaviour for different actions; for example, an action that would require a large amount of concentration in a real person might involve a large number of long gazes at the main target; other targets might be

monitored by occasional glances with the head kept still. In the drinking example, the character must keep its head still while actually drinking, so the appropriate flag must be set by the designer. Once the action has been designed and tagged with the requests these are generated automatically at the start of periods; they do not have to be specified by the user or by the invoking routine. Figures 9 and 10 give examples.

Though this sort of action is useful for a user- controlled character, an autonomous character would require some way of automatically invoking actions with an appropriate target at an appropriate time. Attention is very useful for this as it determines when a character becomes aware of an object and so when it can react to it. We have implemented a method by which a character can react to an approaching object by performing an action on it, for example, catching the ball in Figure 11. A new peripheral vision behavioural controller is added to detect objects with a certain property that the character must react to. The property is defined by a linguistic tag that can be added to an object by the world designer. The new peripheral vision controller can be created automatically given just the property name. This peripheral vision controller will send a request to the attention manager when an object with that property is detected, making the character aware of it. The object is then passed to a behavioural controller which detects whether it is on a collision course with the character. This determines whether the object is approaching the character (this works both for the object moving towards the character or the character moving towards the object). If the object is approaching, it is passed to the action itself, which waits until the object is within a certain distance of the character. When it is, the action will start with the object as a target. Figure 11 gives an example with the character catching an approaching ball. This is a very simple reactive behaviour and is only an initial example of how attention can be used to invoke actions. The general framework of a specialist peripheral vision behavioural controller detecting objects and then the attention manager passing them on to other behavioural controller allow a variety of types of behaviours to be produced. In particular the behavioural controllers dealing with the object could be complex cognitive agents which could use sophisticated methods to decide on an action based on objects that they are aware of. This sort of behaviour gives a wide range of potential further work using our system.



Figure 9: A character putting a ball on a shelf. A piece of motion was transformed to produce this action and gaze behaviour was added by the attention mechanism.

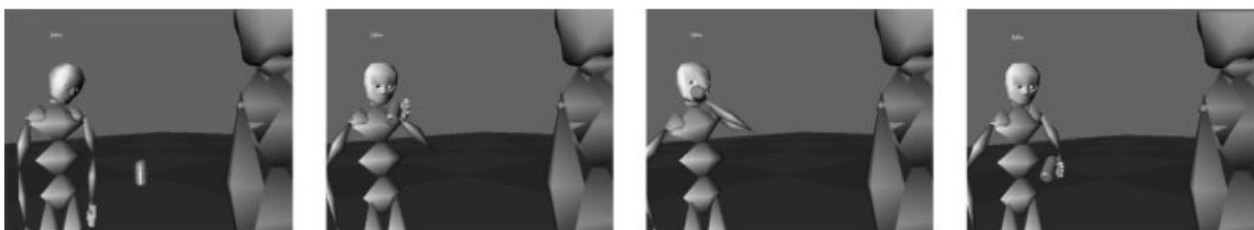


Figure 10: A character drinking from a can. Its eye gaze is mostly downcast, not looking at the other character in the scene, until the last frame, and then without moving its eyes.

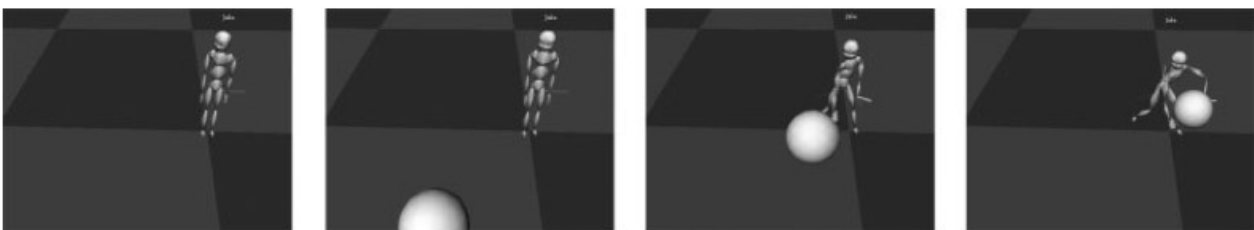


Figure 11: A character reacting to an approaching ball by catching it. The character looks around the environment (frame 1). It then spots the ball and watches it (frame 2). When the ball is close enough the catching is invoked.

Further Work

There are many possible extensions to this work. One of the most important would be to extend the work to include gaze behaviour between characters. The current model is not sufficient to produce the complex interactions of gaze between people.

There are also some other features that would be desirable. It was mentioned above that the length of gaze can be made to depend on the target. It would be desirable to add new features of the target that influence this; in particular a method could be included by which the user can add new dependencies and so alter the character's behaviour. This would be particularly useful when dealing with interactions between characters; altering the length of gaze that a character gives to a particular person can have a wide range of interpersonal meanings.

The user can influence the character's gaze behaviour through a number of parameters which control the attention manager and the behavioural controllers producing requests. Currently this is done using a set of sliders. This method could be greatly improved by adding a good high-level interface for controlling these parameters, in particular mapping low-level parameters onto more intuitive high-level features.

Finally, at present the architecture does not have a system of priorities between gaze requests. For example, it would be useful if a higher-priority immediate request could take the lock from a lower-priority one. This sort of low-level feature can be added but a priority system would have to be mostly implemented in terms of behavioural controllers, not attention requests, as it is the importance of the individual behaviours that determines whether a request should preempt another. At present our behaviours are such that they do not have a natural priority model.

Conclusion

We have presented a general system for producing attention behaviour in a computer-animated character. This system has a number of features:

- A variety of different types of attention behaviour are available for different contexts. Immediate shifts and monitoring fulfil the requirements of different contexts. Undirected attention is for the context where no specific attention behaviour is required. However, it is still adaptable to different contexts and characters by changing the default direction, the probability of looking around, or the interest values of the various objects in the scene.
- The character's attention behaviour is dependent on what other actions the character is performing. The behavioural controllers that are controlling the other actions can request different types of attention behaviour. They can use different ways of specifying a target: as an object, a location or a direction. There are also a number of parameters that the behavioural controllers can use to control how the attention behaviour is performed.
- The user or other behavioural controllers can control the way in which the character performs its attention behaviour as a whole using various parameters. These can change the character of the behaviour. * The attention mechanism can be used by other behavioural controllers to determine what the character is attending to or aware of. This can be used to create behavioural controllers that depend on attention.

Acknowledgments

We would like to thank the UK Engineering and Physical Sciences Research Council for funding this research, the Cambridge University Rainbow Research Group for support and advice, BTextact Technologies for assistance with writing this article and anonymous referees for useful comments.

References

1. Garau M, Slater M, Bee S, Sasse MA. The Impact of eye gaze on communication using humanoid avatars. In ACM SIGCHI, March 31–April 5, Seattle, WA; ACM Press, 2001; pp 309–316.
2. Pashler H (ed.). Attention. Psychology Press: Hove, UK, 1998.
3. Chopra-Khullar S, Badler N. Where to look? Automating visual attending behaviors of virtual human characters. In Autonomous Agents Conference, May 1–5, Seattle, WA; ACM Press, 1999.
4. Thórisson K. Real-time decision making in multimodal face-to-face communication. In Second ACM International Conference on Autonomous Agents, May 9–13, Minneapolis, St Paul; ACM Press, 1998; pp 16–23.
5. Vilhjálmsdóttir HH, Cassell J. Bodychat: autonomous communicative behaviors in avatars. In Second ACM International Conference on Autonomous Agents, May 9–13, Minneapolis, St Paul; ACM Press, 1998.
6. Colburn A, Cohen M, Drucker S. The role of eye gaze in avatar mediated conversational interfaces, Technical report, Microsoft Research, 2000.
7. Hill RW. Modelling perceptual attention in virtual humans. In 8th Conference on Computer Generated Forces and Behavioural Representation, 1999.
8. Rickel J, Johnson WL. Animated agents for procedural training in virtual reality: perception, cognition, and motor control. Applied Artificial Intelligence 1999; 13: 343–382.
9. Argyle M, Cook M. Gaze and Mutual Gaze. Cambridge University Press: Cambridge, UK, 1976.

10. Yarbus AL. Eye Movements and Vision. Plenum Press: New York, 1967.
11. Kahneman D. Attention and Effort. Prentice-Hall: Englewood Cliffs, NJ, 1973.
12. Lee DN. Visuo-motor coordination in space time. In *Tutorials in Motor Behavior*, Stelmach GE, Requin J (eds). *Advances in Psychology*. North-Holland: Amsterdam, 1980; pp 281–295.