Volume xx (*200y*), *Number z, pp. 1–9*

Synthetic Vision and Memory for Autonomous Virtual Humans

C. Peters and C. O' Sullivan[1]

[1] Image Synthesis Group, Department of Computer Science, Trinity College, Dublin 2, Republic of Ireland

Abstract

A memory model based on "stage theory", an influential concept of memory from the field of cognitive psychology, is presented for application to autonomous virtual humans. The virtual human senses external stimuli through a synthetic vision system. The vision system incorporates multiple modes of vision in order to accommodate a perceptual attention approach. The memory model is used to store perceived and attended object information at different stages in a filtering process. The methods outlined in this paper have applications in any area where simulation-based agents are used: training, entertainment, ergonomics and military simulations to name but a few.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Virtual reality

1. Introduction

When modelling agent-object interactions in virtual environments, virtual humans are generally provided with complete access to all objects in the environment, including their precise current states, through the scene database. This is conceptually unrealistic - as real humans we know that life is not that simple. When we are getting dressed in the morning and need to find the companion to that sock underneath our bed, we do not have the luxury of requesting its whereabouts from a scene database. Instead, we must use our intelligence, knowledge and senses to find it. Obviously, our memory of everything from what a sock looks like, to where we usually keep socks, plays a key role in the process.

If we agree that endowing an agent with the ability to follow this process would improve their level of autonomy, it may also be agreed upon that the purpose of the search is not only important in terms of functionality for the agent that initiates it, but perhaps also just as important in terms of plausibility from the point of view of those who witness it.

Given that an agent is made autonomous in this way, we must then equip that agent with the ability to store useful data and disregard extraneous information. Luckily, it turns out that real humans have a very elaborate system for doing this already. Using perceptual attention, we limit our processing to restricted regions of interest in our environment in order to balance the scales between perception and cognition.

This paper combines a synthetic vision module with a memory model based on stage theory[2] to provide a virtual human with a means of attending to their environment. Attention is very important with respect to memory, since it can act as a filter for determining what information is stored in memory and for how long. We focus on goal driven attention as opposed to stimulus driven attention, since the methods described here are intended for use in an autonomous prehension system.

This article is structured as follows: Section 2 looks at the background of the computer animation and puts our character animation research into context with respect to the field of computer animation. We categorise our method of animation in terms of autonomy and interactivity. Section 3 reviews work related to our research in a more in-depth manner. Section 4 introduces our method of synthetic vision and shows how it provides a fast and simple method of sensing for our characters. Section 5 looks at our memory model and why it is an important ingredient in our animation control process. Section 6 discusses the current implementation of the interactions between the vision and memory system. Section 7 focuses on work in progress, more specifically, on improving the synthetic vision system and the long-term memory

module. Section 8 is entitled 'conclusions and future work' and details a more elaborate system for controlling the filtering process and driving the vision system, based on a model of human attention.

2. Background

Human character animation is an intriguing and challenging aspect of computer animation. It is also undoubtedly a highly important aspect: human characters are widely used in productions ranging from video games to animated films. The challenge is two-fold: first, to provide a way to control a highly complex structure comprising many joints, subject to kinematic or dynamic constraints; second, to produce body poses and motions that appear to be natural. This is made considerably more difficult by the fact that the observers, as human beings, could be considered experts when viewing the motions of other humans. Although we may not always be able to articulate a precise problem with a motion, we may be left with a feeling that something is wrong. Subtleties are often critical in such situations.

There are a number of methods available for animating human characters. The use of such methods is limited by their intended application. Generally speaking, these methods differ in terms of the degree of autonomy they provide, and the degree of interactively they allow. In terms of autonomy, some characters are completely user controlled, and require a human operator to specify all joint motions by hand. While providing plenty of control, such an approach can prove tedious for complicated hierarchical characters. At the other extreme are characters that are completely autonomous and do not require the intervention of a user at all. Here, all motions are controlled entirely by a software program. Here, the drawback is that it may be difficult to get a character to act in a certain way. Interactivity is also important.

Interactivity can be thought of as the time taken for character to react to an action. Highly-interactive characters may be interacted with in real-time and calculations for such characters have strict execution rates. In contrast, non-interactive characters often have calculations that are conducted off-line. In this paper, we are interested in the highly-interactive animation of autonomous characters; that is, characters should be able to plan their own motions and the time expended on the calculation of these motions should not be excessive.

We categorise our approach to the problem of human character animation as behavioural animation. In behavioural animation, a character determines its own actions to a certain extent. This gives the character the ability to deal with dynamic environments or unforeseen circumstances. Most importantly, it seeks to free the animator from the need to specify every detail of a character's animation. In order for a software program to control a character, it must have some sort of model of behaviour for that character. As noted in Renault et al[13], behaviour is often defined as "the way that animals and human beings act" and is also often reduced to reaction to the environment. We agree with their statement that a better definition should also include:

> "the flow of information by which the environment acts on the living creature as well as the ways the creature codes and uses this information"

It is further noted by Gilies[7] that in order for a simulation of human behaviour to be effective "it must include the characters' interaction with their environment and to do this it must simulate the characters' perception of the environment". As will be seen in section 3, the behavioural animation approach has been very successful at providing animations for simpler organisms such as schools of fish and flocks of birds. Unfortunately, human behaviour is not easy to model effectively. The human being is exceptionally complex, and many mental processes are not well understood.

3. Related Work

A number of researchers have studied the endowment of agents with internal sensory and storage mechanisms for the purposes of animation. Early research proved particularly successful at animating animal behaviour.

Reynolds[16] presents a distributed behavioural model for flocks of birds and herds of animals. The method is based on the insight that elements of the real system (birds) do not have complete and perfect information about the world and that these imperfections have a major impact on the final behaviour of the system. The system is based on simulated birds, or boids. These are similar in nature to the individual particles in a particle system. Each boid is implemented as an independent actor that navigates according to, among other things, its local perception of the dynamic environment. The boid model does not attempt to directly simulate the senses used by the real animals; rather it attempts to make the same final information available to the behavioural model that the real animal would receive as an end result of its perceptual and cognitive processing. Each boid has a spherical zone of sensitivity centered at its local origin, and the behaviours that comprise the flocking model are stated in terms of nearby flock-mates. A key issue is raised here regarding behavioural animation: how to analyse the success of the model. As Reynolds notes, it is difficult to objectively measure how valid such simulations are. However, the flocks built from the model seem to correspond to the observer's intuitive notion of a 'flock-like motion'. An interesting result of the experiments with the system reveal that the 'flocking' behaviour that we intuitively recognise is not only improved by a limited, localized view of the world, but is dependent on it.

Tu et al[17] present a framework for animation featuring the realistic appearance, movement and behaviour of individual and schools of fish with minimal input from the animator.

Their repertoire of behaviours relies on their perception of the dynamic environment. Individual fish have motivations as well as simple reactive behaviour. At each time step, habit, mental state and sensory information are used to provide an intention. Behaviour routines are then executed based on this intention and motor controllers provide motions that fulfil these behaviours. Habits are represented as numerical variables for determining individual tendencies towards brightness, darkness, cold, warmth, and schooling. The individual fish also has three mental state variables for hunger, libido and fear. Behaviour patterns may be interrupted by reactions to more pressing environmental stimuli (for example, a predator). The artificial fish has two on-board sensors with which to perceive its environment - a vision sensor and a temperature sensor. The vision sensor is cyclopean covering a 300-degree spherical angle. An object is seen if any part of it enters this view volume and is not fully occluded by another object. The vision sensor has access to the geometry, material property and illumination information that is available to the graphics pipeline and also to the object database and physical simulator for information such as identification and velocities of objects. It is noted that as their holistic computational model exceeds a certain level of physical, motor, perceptual and behavioural sophistication, the agent's range of functionality broadens due to emergent behaviours.

Renault et al[13] introduce a synthetic vision system for the high level animation of actors. The goal of the vision system in this case is to allow the actor to move along a corridor avoiding objects and other synthetic actors. For the vision system, the scene is rendered from the point of view of the actor and the output is stored in a 2D array. Objects in the scene are not rendered using their usual colours, but are rendered using unique colours for each object. Each element in the 2D array consists of a vector containing the pixel at that point, the distance from the actor's eye to the pixel and an object identifier of any object that is at that position. The size of the array is chosen so as to provide acceptable accuracy without consuming too much CPU time. A view resolution of 30x30 was selected for the corridor problem.

Noser et al[12] extend previous work[13] by adding memory and learning mechanisms. They consider the navigation problem as being comprised of two parts: global navigation and local navigation. Global navigation uses a simplified map to perform high-level path-planning. This map is somewhat simplified, however, and may not reflect recent changes. In order to deal with this, the local navigation algorithm uses direct input from the environment to reach goals and sub-goals given by the global navigation systems and to avoid unexpected obstacles. This local navigation algorithm has no model of the environment and does not know the position of the actor in the world. The scene is rendered as before and global distances to objects are extracted for use by the navigation system. An octree data structure for the 3D environment is constructed from the 2D image and the depth information. This data structure represents an actor's long-term visual memory of the 3D environment, and can handle static and dynamic objects. Using this long-term memory, an actor can find 3D paths through the environment avoiding impasses.

Kuffner et al[10] present a perception-based navigation system for animated characters. Of particular interest to this study is an algorithm for simulating the visual perception and memory of a character. The visual system provides a feedback loop to the overall navigation strategy. The approach taken builds on previous approaches[12]. An unlit model of the scene is rendered from the characters point of view using a unique colour assigned to each object or object part. These objects and their locations are added to the character's internal model of the environment. A record of perceived objects and their locations is kept as the character explores an unknown virtual environment, thus providing a kind of spatial memory for each character. Unlike previous approaches, this method relies on the object geometry stored in the environment and a list of objects IDs and positions. This provides a relatively compact and fast representation of each character's internal world. Previously unobserved objects are added to the characters list of known objects, and other visible objects are updated with their current transformation. Objects that were previously visible but are no longer in view retain their most recent observed transformations.

Blumberg[3] presents an ethologically inspired approach to real-time obstacle avoidance and navigation. Again, a creature renders the scene from its own viewpoint. This rendering is used to recover a gross measure of motion energy as well as other features of the environment, which are then used to guide movement. An approximate measure of motion energy is calculated for each half of the image, which is then used to provide corridor following and obstacle avoidance.

Gillies[7] uses a psychological approach to design a visual algorithm. The methods do not try to simulate the image on the retina of the actor or allow the actor to perceive features such as colour and shape. Rather, they work at a higher level, using basic features such as velocity and position to calculate object features. Object features are rather abstract and represent complex reasons as to why an object might be looked at. Interest would be an example of an object feature. The system does not attempt to provide meaning for object properties and actors show more interest in some properties than in others. This means that the actors will have different reactions to an object.

Chopra et al[5] propose a framework for generating visual attention behaviour in a simulated human agent based on observations from psychology, human factors and computer vision. A number of behaviours are described, including eye behaviours for locomotion, monitoring, reaching, visual search and free viewing.

Hill[8] provides a model of perceptual attention in order

to create plausible virtual human pilots for military simulations. Objects are grouped according to various criteria, such as object type. The granularity of object perception is then based on the attention level and goals of the pilot.

4. Synthetic Vision

Synthetic senses provide the means for actors to perceive their environment through an indirect means. We regard such methods as being indirect, since actors are normally unrestricted when performing interrogations of the environment's state in the database. Although this direct access method is simple and fast, it suffers from scalability and realism problems. Because of this, research has focused on providing agents with their own methods of perceiving the environment. In some cases, actors are provided with a sensory sphere that may deform in a direction depending on the velocity in that direction[16]. Although this approximation is adequate to produce realistic group behaviour, it is mentioned that individuals would be better at path planning if they could see their environment. Indeed, it has been noted that most characters do not have an omni-directional perception; sensory information from the environment flows from a primary direction, such as the cone of vision for a human character[9].

We focus on the visual modality of sensing in this paper. Vision is regarded as the most important of all the senses for humans. Research on synthetic sensors for other modalities has also been conducted[14].

It should be noted that the aim of the vision approach described here is not necessarily to imitate the human visual system as accurately as possible. Instead, it is to provide a reasonable estimate of what the visual system senses without incurring the costs associated with simulating the highly complicated mechanisms of the eye. For example, our system is monocular since object depth information may be obtained during the rendering process. Noser et al[14] differentiate between synthetic vision and artificial vision. While synthetic vision is simulated vision for a digital actor, artificial vision is the process of recognising the image of a real environment captured by a camera. Since artificial vision must obtain all of its information from the vision sensor, the task becomes more difficult, involving time-consuming tasks such as image segmentation, recognition and interpretation.

There are a number of reasons for adopting a computer vision technique. First of all, it may be the simplest and fastest way to extract useful information from the environment[3]. Underlying hardware can be taken advantage of and, since the object visibility calculation is fundamentally a rendering operation, all of the techniques that have been developed to speed up the rendering of large scenes can be adopted. These include scene-graph management and caching, hierarchical level-of-detail (LOD) and frame-to-frame coherency[9]. Secondly, synthetic vision may scale better than other techniques in complex environments. By its very nature, the visual system provides a controllable filtering mechanism so as not to overwhelm our limited cognitive abilities. Thirdly, the approach makes the actor less dependent on the underlying implementation of the environment because it is not necessary to rely directly on the scene database. Finally, Blumberg[3] notes "... believable behaviour begins with believable perception". There are also some beneficial side effects to this method: occluded objects are implicitly handled in a static scene and the method may be extended to dynamic scenes through the use of a memory model.

Our synthetic vision module is based on the model described by Noser et al[12]. This model uses false-colouring and dynamic octrees to represent the visual memory of the character. We adopt a similar system to Kuffner et al[10], by removing the octree structure. Rather, scene description information is encoded with a vector that contains object observation information.

The process is as follows: Each object in the scene is assigned a single, false colour. The rendering hardware is then used to render the scene from the perspective of each agent. The frequency of this rendering may be varied. In this mode, objects are rendered with flat shading in the chosen false-colour. No textures or other effects are applied. The agent's viewpoint does not need to be rendered into a particularly large area: our current implementation uses 128x128 renderings (See Figure 2). The false-coloured rendering is then scanned, and the object false-colours are extracted.

We extend the synthetic vision module by providing multiple vision modes. Each mode uses a different palette for false-colouring the objects. The differing vision modes are useful for capturing varying levels of information detail of information about the environment. The two main vision modes are referred to as *distinct mode* and *grouped mode*.

In the *distinct vision mode*, each object is false-coloured with a unique colour. The unique colours of objects in the viewpoint rendering may then be used to do a look-up of the object's globally unique identifier in the scene database. This identifiers is then passed to the memory model. This mode is useful when a specific object is being attended to (Figure 2).

The other primary vision mode is called *grouped vision mode*. In this mode, objects are false-coloured with group colours, rather than individual colours. Objects may be grouped according to a number of different criteria. Some examples of possible groupings are brightness, luminance, shape, proximity and type. The grouped vision mode is useful for lower detail scene perception (Figure 2). Note that the grouped vision mode only provides information about potentially visible objects. It is entirely possible that a group will be marked as being in view when only one of the objects in the group are actually in view. For example, consider a group consisting of a table with numerous glasses and a large bottle on it. Although the table and glasses may be out of view, the

ObjID	globally unique identifier of the object
objAzi	azimuth of the object
objEle	elevation of the object
objDis	distance to the object
t	time-stamp

Table 1: *Representation of observations*

group will still be tagged as potentially visible, since part of the bottle may be in view. Essentially, this means that further querying is required on the members of a group to establish if they are in view.

The information acquired by the virtual human under the above circumstances is referred to as an observation. In our implementation, the precise position of an object or group in the environment is not stored as part of an observation unless a certain amount of attention has been given to it. Rather, an approximation of the object's location in spherical coordinates with respect to the agent's viewing frame is used. During the scanning process, bounding boxes are assembled for each object based on the object's minimum and maximum x and y coordinates extracted from the view specific rendering and the object's minimum and maximum z coordinates extracted from the z-buffer for that view. The object's position is then estimated to be the centre of this bounding box. This process has the overall effect of making accurate judgements about the positions of partially occluded objects more difficult. Also, estimates made about the distance to the centre of the object will vary depending on the obliqueness of the object with respect to the viewer.

An observation is represented as a tuple that is composed of five components, shown in Table 1. A specific object will have at most a single observation per agent. The observation will match the last perceived state of the object, although it must be noted that this may not correspond to the actual current state of the object. Observations are also stored for groups of objects, using a similar process, where groups are bounded and their positions calculated as above. Finally, it should be noted that when observations are stored as memories (see Section 5), their coordinates are expressed in Cartesian rather than spherical coordinates.

5. Memory Model

Some form of memory is crucial for agents that are disconnected from the environment database in some way. As with living creatures, autonomous agents rely on their memory to differentiate between what they have and have not observed. An agent that automatically knows the location of every object in the scene will destroy its plausibility with respect to a human viewer, while an agent that has no memory of its surroundings will appear to be stupid when conducting tasks. There are many instances where memory plays a large part in everyday human behaviour. Consider the example of searching for an item that is out of view. In the case where an item has been seen previously and remembered, the search task can begin with the remembered position of the object. In contrast, one who had no memory of the position of the item would be obliged to embark on a lengthy search. Such imperfections (for example, having to search for an item) are part of everyday human life and, as such, their absence may be a factor in decreasing the plausibility of an agent's actions with respect to a viewer. Memory is a way of deciding when such actions are necessary.

We apply research from the field of cognitive psychology in order to provide a simplified model of memory. Despite decades of research on the subject, human memory still provides many great challenges to cognitive psychologists and there are many areas of lively debate. As with the synthetic vision model described in the previous section, we seek to learn high-level lessons from research on the real system, and use this to create a simplified model that will suit our needs. In this case our needs are two-fold: create a memory mechanism for the agent while minimising the storage requirements.

One of the main arguments against incorporating independent memory systems appears to be that of storage. It has been suggested that a memory model entails the storage of multiple copies of the world database[7]. In the worst case, this would be the same size as the world database for an agent that had perceived every object in the world. This need not be the case, however, if filtering and forgetting mechanisms are applied to the memory system. Filtering makes sure that only important information makes it through to the storage stages, while forgetting clears out information that is no longer as valuable as it used to be. In fact, a plausible argument is that this filtering and forgetting is what allows real humans to cope with the huge amounts of data that they sense[8]. In our framework, synthetic vision is just the first in a number of filters that reduce the amount of data that must be stored and processed by each agent. Memory is implemented as the main filtering mechanism; only information that is deemed to be important will be stored for any length of time. In this section, we will discuss only the structure and basic operations of the memory model that is being used. An in-depth discussion of the controller of both the vision and memory modules is outside of the scope of the paper; it is work in progress and is discussed in Section 6.

Over the past 40 years, a number of different structural analyses of memory have been performed. These have been conducted on normal individuals as well as individuals suffering from brain damage and disease. We base our system of memory on what is referred to as *stage theory*[2]. They propose a model where information is processed and stored in 3 stages: sensory memory (STSS), short-term memory (STM) and long-term memory (LTM). See Figure 1 for a schematic of the model. This model provides a useful structure.

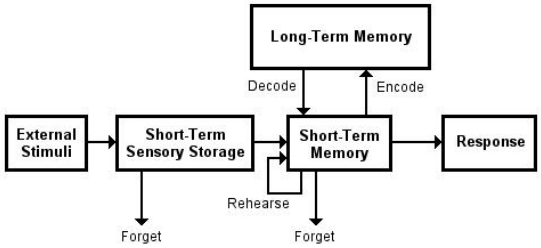

Figure 1: *Schematic of the memory model from* [2].

Short-term sensory storage (STSS) is a short duration memory area where a variety of sources of information (e.g. light, smell, sound, etc) are converted into signals that the brain can understand. Since this memory has a very fast rate of decay, it is essential that information be attended to in order to transfer it to the next stage of processing (short-term memory). Our model of STSS only takes account of the visual modality and is derived from the viewpoint rendering discussed previously. Observations extracted from this rendering comprise the STSS. We allow a large number of observations to be stored in the STSS, although it should be noted that only visually sensed items will make it into this memory and many of these items will be groups of objects rather than individual objects. The STSS is updated with each refresh of the viewpoint rendering.

Short-term memory (STM) relates to our thoughts at any given moment in time. It is created by attention to an external stimulus or internal thoughts. Short-term memory is limited both in duration and by the number of units of information that can be processed at any one time. Research suggests that the STM can process between 7ś2 and 5ś2 units or chunks of information[11]. These units correspond to letters, numbers, and also larger units such as words and phrases.

Our model allows a maximum of 8 units of storage, where we define a unit as either an object observation or a group observation. Memory entries are removed from the STM under two conditions: they are displaced by newer memories when the STM is full and they also decay over time (forgetting). The default time allotted to each memory in the STM module is 20 seconds, after which it decays. In the case where the memory entry is rehearsed however, we extend the time allotted to the memory to 20 minutes. Rehearsal occurs when attention is paid to a specific object over a period of time. In general, we assume that the more an item is attended, the longer it will be allowed to stay in the STM. Because we use a goal-directed attention approach, the items that are attended to (and thus, would be expected to occupy the STM) will be those relating to the goal. Take, for example, the goal of searching for the brown bottle object in a scene. At the end of this search, we would expect the STM to contain other bottles that the agent attended, the group containing the brown bottle, and finally the brown bottle object itself.

Our long-term memory (LTM) gives us the ability to store information long-term, and generally allows this information to be recalled provided suitable cues are available, although it may take several minutes or even hours to do this. We are primarily interested in providing workable answers to two questions regarding long-term memory; the first question is what do we store in our long-term memory, and the second question is how long do we store that material for. In attempting to answer both questions here, we make a number of simplifications. First of all, for the purposes of our demonstration, we assume that only the subject of the task at hand is memorised. That is, if the task is to pick up a glass, then the glass object will be memorised in the agent's LTM as an observation. We also assume that items that are stored in the LTM never decay; that is, they are never forgotten. As such, a more comprehensive long-term memory system is currently the focus of research (see Section 7 for discussion).

Attention is currently modelled in the system by using the different vision modes to control the detail of the information acquired. When the agent becomes attentive towards an object, that object is rendered in the *distinct vision mode* mentioned earlier. In this mode, the full object data may be obtained, including its globally unique identifier. The pre-attentive agent state is modelled using the *group by proximity* vision mode. In this mode, individual objects are not discerned, but rather the states of whole groups of objects are perceived. This type of filtering allows the virtual human, as well as the real human, to reduce large amounts of perceptual data into a manageable size. The *group by type vision mode* could be viewed as being part of the attention acquiring process. It operates with finer granularity than the *group by proximity* mode and is suitable for goal-directed requests by object type (e.g. "take a bottle").

6. Implementation

The implementation of memory is split into a number of separate memory modules: one each for the STSS, STM and LTM. Each memory module is based on memory duration, capacity and a rehearsal value. Unlike the other memory modules, the STSS module also contains the view-port rendering. Each module contains a list of memory entries. A *memory entry* contains an observation and other information such as how many times the memory has been rehearsed and when the last rehearsal took place. The LTM module contains encode (add memory), decode (retrieve memory) and recall (query memory) functions. When an item is retrieved from LTM, it is moved into the STM, overwriting anything currently in the STM. This is useful for modelling a context switch, where the agent's focus of attention is changed.

Our implementation of the goal driven memory and attention process is summarised as follows:

A goal command is given to the virtual human. This goal command contains the globally unique identifier of the object that attention is to be directed towards. If the object is already memorised in the STM or the LTM, then the observation information is extracted and the virtual human will become attentive towards (look at) the object and update its perception of the object using the distinct vision mode. If the object was memorised in the STM, this procedure is regarded as a rehearsal.

If the object is not in the STM or the LTM, then the agent's perception of the environment will be rendered using the *group by proximity* vision mode (currently, agents do not initiate an active search of their surroundings; they only search the groups in their view at the time the task is issued). They then go through the groups in the STM one by one, and render them using the *group by type* vision mode. If an object of the same type as the requested object is there, then they become attentive towards the object and check to see if it is the goal object. If it is not, the search continues through other objects of similar type in the group, and in the case where there are no more, the search proceeds to other groups. If it is the goal object, the perceived state of the object is entered in the STM.

The memory model outlined above was implemented on the ALOHA animation system, an animation system for the real-time rendering of characters[6]. This system uses the OpenGL API on a Windows platform. Figure 3 shows some sample screenshots from the ALOHA system.

7. Work in Progress

One problem with the synthetic vision approach described above is that objects that are sufficiently small or far away with respect to the agent may not be rendered in the agent's view due to its limited resolution; our current implementation uses 128x128 renderings. Our solution to this problem is inspired by lessons from the real system. The retina of the human eye consists of approximately 127 million light-sensitive cells[4]. These are split into cells called rods and cones. The majority of the cells are of the rod variety: they are highly sensitive to light, but are not sensitive to colour. The remainder of the cells are cones. Although there aren't as many cones, they are more sensitive to colour. The cones are concentrated in a circular area known as the macula lutea in the centre of the retina. Within this area is a depression known as the fovea, consisting almost entirely of cones, that spans a visual angle of approximately 2 degrees. Humans use the fovea to make detailed observations about the world. In terms of the approach that we have chosen, the most straightforward analogy to the human fovea is to create a second, higher resolution rendering for a smaller field of view in the scene (a few degrees should suffice). Indeed, a similar approach was used by Tu et al[17], although three renderings, each with increasing resolutions, are taken for each eye. We feel that two renderings is an acceptable compromise between speed and functionality, while at the same time showing parallels to the real-life system.

In terms of the memory model, work is currently focusing on a more elaborate long-term memory system. Long-term memory poses a particular problem to research involving agents. Perhaps the most interesting question is "when do we forget things?" This is also a very important topic of research in the field of cognitive research. Research indicates that there are two main reasons why we may be unable to recall what we have committed to memory. The first is that the memory has disappeared and the second is that the memory is still there but we cannot retrieve it. This can be problematic since it is not always easy for researchers to distinguish between the two possibilities. The most interesting possibility is that we never really lose our memories; all memories are still there, but we cannot retrieve them[1]. Of course, for a system involving agents, we must assume that under some circumstances, memories are forgotten. The goal of the long-term memory should be to store only information that is important to the agent. This seems to suggest both a filtering process that only allows important information in, and a forgetting process that keeps only the most important information. It is likely that the filtering process will be linked to the attention mechanism discussed in section 8. In terms of the forgetting process, Anderson[1] offers some insight:

> "Speed and probability of accessing a memory is determined by its level of activation, which in turn is determined by how frequently and how recently we used the memory."

Forgetting can therefore occur according to a heuristic involving the memory's frequency and recency.

8. Conclusions and Future Work

We have presented a memory model that uses a synthetic vision module in order to acquire information about a virtual environment. The granularity at which this information processed by an agent is determined by the use of multiple vision modes. As mentioned, the intended purpose for the memory model is for the implementation of an attention-based prehension system for virtual humans. Aside from modelling virtual human prehension, work will also focus on a realistic visual search algorithm.

We view the senses, memory and attention as being integral parts of a feedback loop providing realistic motor control (behaviours). Each plays an important role in this functioning: the senses provide us with limited information about our environment; our memory allows us to keep a store of important objects; our attention allows us to decide what objects are more important than others. We believe it is valuable to learn lessons that can be applied to generating more plausible animations from a system that is quite ingenious: the human body.

The main thrust of future research will be on the control

mechanism for deciding what items from the STSS are entered into the STM. The closest parallel to this mechanism in the real human is referred to as *attention* and is an elusive concept. Pashler[15] goes as far as to assume that:

> "... no one knows what attention is, and that there may not even be an "it" there to be known about."

Nonetheless, from our point of view, the concept of an attentional construct is useful. It fits neatly in with the idea of a series of filtering mechanisms for extracting and storing important information from the environment. A more proactive attentional mechanism may also prove useful for providing low-level behavioural animation and establishing a sense of presence; simple orienting behaviours towards important stimuli are a glaring omission when dealing with contemporary autonomous characters.

The output from our attention model will essentially be a list of the current objects in the scene that the agent is aware of (not necessarily all those objects that are visible) and a ranking of these objects based on how interesting they are to the agent. In such a case, specifying an agents *interest* becomes a difficult problem. We approach this problem by viewing interest as being a combination of bottom-up, attention-grabbing processes and top-down, task-related processes. Both are necessary for human survival: for example, while carrying out a task-related attention (looking at your watch to find out the time) you may become aware of a looming object in your periphery of vision (a speeding car). In this case, an attention control process interrupts the task at hand and switches attention towards the more immediate threat.

The above results will be used to provide an object of interest at any one time for the agent, which will invoke higher-level behaviours (for example, an orienting behaviour towards an interesting object). Overt attention may be especially important in providing basic low-level attention behaviours to agents, in order to increase their realism and the viewer's sense of presence.

This article has considered the first steps towards providing attention-driven behavioural animation. These are all internal however. It is envisaged that the results of the work above will be used to create low-level behaviours. Example behaviours that we are interested in include directing character's eye-gaze in stimulus-driven task-dependant situations and using memory and vision to affect search and prehension behaviours.

It should be noted that currently the grouping mechanism for the grouped vision mode is pre-processed; essentially, objects are assigned group ids by human intervention at the start of the program. Such a method will not suit dynamic scenes. A structure for grouping objects according to varying criteria (proximity and type) in a dynamic scene would be complimentary to the current system. This also ignores a more fundamental question: how do humans group objects? More elaborate research on perceptual grouping and how it relates to task requirements would certainly be interesting.

References

1. Anderson, J.R. Cognitive Psychology and Its Implications. 4th edn, New York: Freeman, 1995. 7
2. Atkinson, R. and R. Shiffrin. Human memory: a proposed system and its control processes. In K. Spence and J. Spence editors, the psychology of learning and motivation: advances in research and theory, Vol. 2. New York: Academic Press, 1968. 1, 5, 6
3. Blumberg, B. Old Tricks, New Dogs: Ethology and Interactive Creatures. PhD Dissertation, MIT Media Lab, 1996. 3, 4
4. Bruce, V. and P.R. Green. Visual Perception: physiology, psychology and ecology. 2nd edn, Lawrence Erlbaum Associates Ltd., Hove, U.K., 1990. 7
5. Chopra, S. and N. Badler. Where to look? Automating attending behaviours of virtual human characters. Autonomous Agents and Multi-Agent Systems. 4 (1/2):9-23, 2001. 3
6. Giang, T., R. Mooney, C. Peters, and C. O'Sullivan. ALOHA: adaptive level of detail for human animation. Eurographics 2000, Short Presentations, 2000. 7
7. Gillies, M. Practical Behavioural Animation Based On Vision Asnd Attention. University of Cambridge Computer Laboratory, Technical Report TR522, 2001. 2, 3, 5
8. Hill, R.W. Perceptual Attention in Virtual Humans: Towards Realistic and Believable Gaze Behaviours. Simulating Human Agents, Fall Symposium, 2000. 3, 5
9. Kuffner, J. Autonomous Agents for Real-Time Animation. PhD Dissertation, Stanford University, 1999. 4
10. Kuffner, J. and J.C. Latombe. Perception-Based Navigation for Animated Characters in Real-Time Virtual Environments. The Visual Computer: Real-Time Virtual Worlds, 1999. 3, 4
11. Miller, G.A. The magic number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, pages 81-97, 1956. 6
12. Noser, N., O. Renault, D. Thalmann, and N.M. Thalmann. Navigation for digital actors based on synthetic vision, memory and learning. Computer and Graphics, Vol. 19, pages 7-19, 1995. 3, 4
13. Renault, O., D. Thalmann, and N.M. Thalmann. A vision-based approach to behavioural animation. Visualization and Computer Animation, Vol. 1, pages 18-21, 1990. 2, 3

14. Noser, N. and D. Thalmann. Synthetic Vision and Audition for Digital Actors. Proc. Eurographics '95, Maastricht, pages 325-336, 1995. 4

15. Pashler, H.E. The Psychology of Attention. Cambridge, Massachusetts, MIT Press, 1998. 8

16. Reynolds, C.W. Flocks, herds and schools: A distributed behavioural model. Computer Graphics, 21(4), pages 25-34, 1987. 2, 4

17. Tu, X. and D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behaviour. Proc. SIGGRAPH '94, pages 43-50, 1994. 2, 7

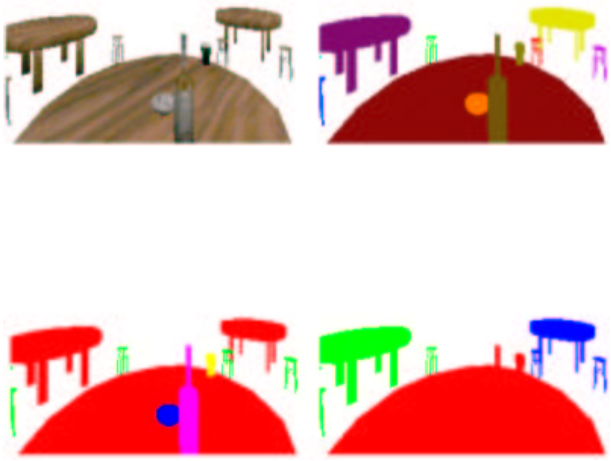

Figure 2: *Sample object views as seen from the perspective of the agent with (a) no false colouring applied, (b) false colouring according to object id, (c) false colouring according to object type, and (d) false colouring according to object proximity.*

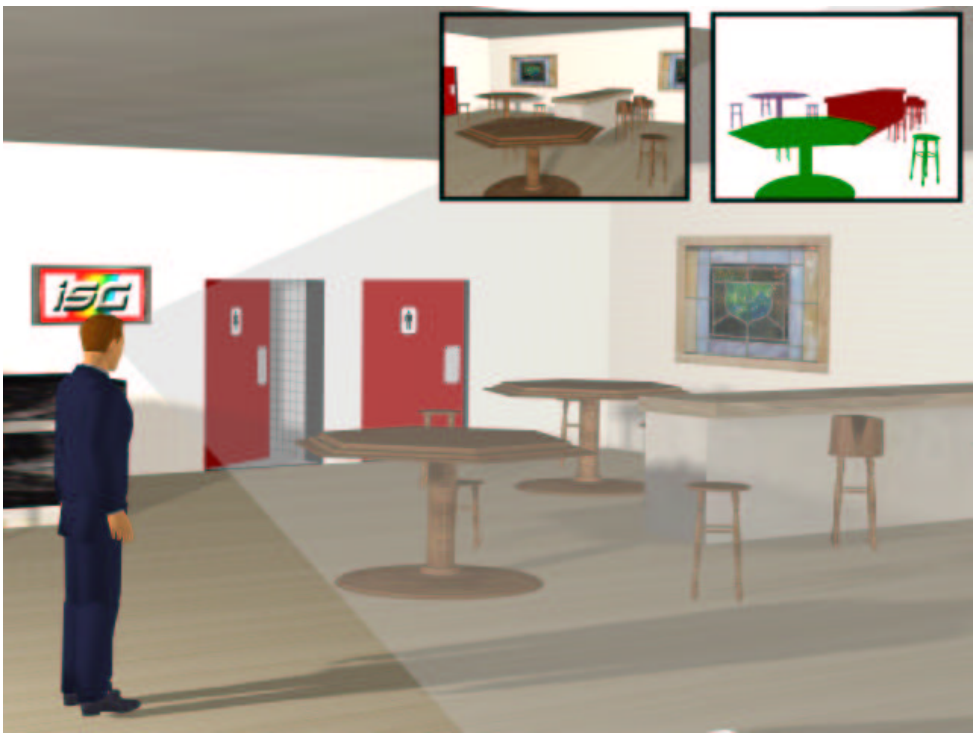

Figure 3: *The system in action. The left inset shows the scene rendered from the viewpoint of the agent. The right inset depicts the same view false-coloured according to object proximity.*