January 1999

Where to Look? Automating Certain Visual Attending Behaviors of Human Characters

Sonu Chopra-Khullar

University of Pennsylvania

Follow this and additional works at: http://repository.upenn.edu/ircs_reports

http://repository.upenn.edu/ircs_reports/41


This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ircs_reports/41
For more information, please contact libraryrepository@pobox.upenn.edu.
Where to Look? Automating Certain Visual Attending Behaviors of Human Characters

Abstract
This thesis proposes a computational framework for generating visual attending behavior in an embodied simulated human agent. Such behaviors directly control eye and head motions, and guide other actions such as locomotion and reach. The implementation of these concepts, referred to as the AVA, draws on empirical and qualitative observations known from psychology, human factors and computer vision. Deliberate behaviors, the analogs of scanpaths in visual psychology, compete with involuntary attention capture and lapses into idling or free viewing. For efficiency, the embodied agent is assumed to have access to certain properties of the 3D world (scene graph) stored in the graphical environment. When information about a task is known, the scene graph is queried. When an agent lapses into free viewing or idling, no task constraints are active so a simplified image analysis technique is employed to select potential directions of interest. Insights provided by implementing this framework are: a defined set of parameters that impact the observable effects of attention, a defined vocabulary of looking behaviors for certain motor and cognitive activity, a defined hierarchy of three levels of eye behavior (endogenous, exogenous and idling) and a proposed method of how these types interact, a technique of modifying motor activity based on visual inputs, and a technique that allows for anticipation and interleaving of eye behaviors for sequential motor actions. AVA generated behavior is emergent and responds to environment context and dynamics. Further, this method animates behavior at interactive rates. Experiments supporting several combinations of environment and attending conditions are demonstrated, followed by a discussion of an evaluation of AVA effectiveness.

Comments

This thesis or dissertation is available at ScholarlyCommons: http://repository.upenn.edu/ircs_reports/41
WHERE TO LOOK? AUTOMATING CERTAIN VISUAL ATTENDING BEHAVIORS OF HUMAN CHARACTERS

Sonu Chopra-Khullar

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

1999
Acknowledgements

My husband, Rajesh Khullar, has been a great source of kindness, patience and encouragement during the course of this work. I owe thanks to my advisor, Dr. Norman I. Badler, whose unwavering confidence in and tremendous enthusiasm for this research have been of great value. I also owe thanks to Drs. Mark Steedman and Bonnie Webber who, even in the early days of the admission process, had faith in my abilities and gave me the support that made my graduate career possible.
Sponsors

This research has been partially supported by U.S. Air Force through Delivery Order #8 on F41624-97-D-5002; Office of Naval Research (through Univ. of Houston) K-5-55043/3916-1552793; DARPA SB-MDA-97-2951001; NSF IRI95-04372; NASA NRA NAG 5-3990; and JustSystem Japan.
ABSTRACT

WHERE TO LOOK? AUTOMATING CERTAIN VISUAL ATTENDING BEHAVIORS OF HUMAN CHARACTERS

Sonu Chopra-Khullar
Norman I. Badler

This thesis proposes a computational framework for generating visual attending behavior in an embodied simulated human agent. Such behaviors directly control eye and head motions, and guide other actions such as locomotion and reach. The implementation of these concepts, referred to as the AVA, draws on empirical and qualitative observations known from psychology, human factors and computer vision. Deliberate behaviors, the analogs of scanpaths in visual psychology, compete with involuntary attention capture and lapses into idling or free viewing. For efficiency, the embodied agent is assumed to have access to certain properties of the 3D world (scene graph) stored in the graphical environment. When information about a task is known, the scene graph is queried. When an agent lapses into free viewing or idling, no task constraints are active so a simplified image analysis technique is employed to select potential directions of interest. Insights provided by implementing this framework are: a defined set of parameters that impact the observable effects of attention, a defined vocabulary of looking behaviors for certain motor and cognitive activity, a defined hierarchy of three levels of eye behavior (endogenous, exogenous and idling)
and a proposed method of how these types interact, a technique of modifying motor activity based on visual inputs, and a technique that allows for anticipation and interleaving of eye behaviors for sequential motor actions. AVA generated behavior is emergent and responds to environment context and dynamics. Further, this method animates behavior at interactive rates. Experiments supporting several combinations of environment and attending conditions are demonstrated, followed by a discussion of an evaluation of AVA effectiveness.
# Contents

Acknowledgements iii

Sponsors iv

Abstract v

1 Introduction 1

1.1 Background on Eye Movements and Scanpaths Implemented in the AVA Method 4
1.2 Overview of the AVA Method 6
1.3 Utilizing the Graphics Database – Assumptions of the Model 7
1.4 Motivation and Applications 8
1.5 Generating Behavior Versus Generating Plans 10
1.6 Scope of Potential Scenarios 11
1.7 Requirements of the Human Model 12
1.8 Outline of Thesis 13

2 Related Work 14

2.0.1 Cog 15
2.0.2 Conversational Agents, Social Interaction and Intelligent Tutors 16
2.0.3 Facial Animation Systems 17
2.0.4 Motion Capture Techniques 17

vii
2.0.5 Image Processing Approaches ....................... 18
2.0.6 Other Related Techniques ......................... 18

3 Related Psychology Experiments - Basis for the Method 20
3.1 Patterns of Looking Associated with Cognitive and Motor Activities 23

4 AVA Architecture 24
4.1 Properties of Finite State Machine Language .................. 29

5 Eye Behaviors in the AVA 32
5.1 Visual Search ........................................ 32
5.2 Monitoring and Locomotion ............................... 36
5.2.1 Limit Monitoring .................................. 38
5.3 Reaching and Grasp .................................... 39
5.4 Visual Tracking ....................................... 40
5.5 Motion in the Periphery ................................. 41
5.6 Spontaneous Looking .................................. 42

6 Managing the Task Queue and Motor Control 64
6.1 Interleaving and Confidence Levels ...................... 64
6.2 Task Queue Manager and Examples ..................... 65
6.3 Head-Eye Movement and Motor Control .................. 66

7 Worked Examples 70
7.1 Visually Guided Locomotion ............................... 70
7.2 Crossing the Road Example .............................. 74
7.2.1 Details of Simulation ............................... 75
7.2.2 Analysis of Simulation .............................. 76

8 Conclusions 86
8.1 Contribution .......................................... 86
8.2 Limitations .................................................. 90
8.3 Extensions .................................................. 91
8.4 Metrics for Evaluating The AVA .......................... 92
  8.4.1 Generated Behavior and Viewer Beliefs .......... 92
  8.4.2 An Empirical Approach ......................... 93
  8.4.3 Summary of Metrics ........................... 95

A Modification of Data Structures .......................... 96

B Head Eye Coordination Algorithm ....................... 98

Bibliography .................................................. 98
List of Tables

4.1 Eye Behavior Nets ........................................... 31
List of Figures

1.1 Overview of AVA Flow of Control ........................................ 7
4.1 Method Architecture .................................................. 25
4.2 GazeNet ............................................................... 27
4.3 GazeNet Algorithm ..................................................... 30
5.1 Visibility Checking ..................................................... 35
5.2 Looking for the Candy Cane ............................................ 37
5.3 Change Heading if All Rays Unoccluded ............................... 38
5.4 Agent Scanning Environment for the ball – Visibility Rays Unoccluded 46
5.5 Agent Changes Heading and Continues Search ...................... 47
5.6 Agent Searching for and Finding the ball ............................. 48
5.7 Glancing At Goal of Action ............................................. 49
5.8 Visual Tracking – Prediction Past Occlusions ....................... 50
5.9 Agent Assigned Ball and Car Tracking Tasks ......................... 51
5.10 Agent Tracks Car and Attempts to Track Ball ...................... 52
5.11 Agent Follows Car and Subsequently Regains Ball Target ........ 53
5.12 Agent notices moving car and estimates collision likelihood ...... 54
5.13 Agent Resumes Walking along Original Heading .................. 55
5.14 Spontaneous Looking - Rays Intersect Features with Local Contrast . 56
5.15 Spontaneous Looking .................................................. 57
5.16 Spontaneous Looking .................................................. 58
Chapter 1

Introduction

The goal of this research, known as the AVA (Automated Visual Attending), is to provide a computational framework that generates convincing visual attending or looking behaviors for virtual humans in a variety of circumstances. Scenarios which may be generated from this technique are created by combining a supported set of motor and cognitive actions. The resulting behavior is emergent and responds to environment dynamics. Given a loose outline of activity, which may be a high level story board provided by an animator (even entered interactively) or a set of directives from a task planner, this work attempts to generate details of the script with the appropriate looking behavior as well as realizing the corresponding task motions. Motor actions in this system are implemented using the Jack human model’s repertoire of capabilities.

Synthetic humans should exhibit the appropriate looking or attending behaviors relevant to the activities they are engaged in or even in the absence of deliberate intent. Since gaze is significant in communication and behavioral representation, random or uncontrolled looking behavior is both misleading and disconcerting. The naturalness of looking behavior is taken for granted in daily interaction, yet eye movements that fail to to follow a natural pattern are immediately noticeable and, in fact, often an indication of mental disorder. [Sweeney, Clementz, Haas, Escobar,
Drake, and Frances, 1994]. Characters for which motion alone is animated while gaze remains fixed appear robot-like or mechanical. Also, real-time and unscripted performance is necessary for interactive simulation, games and 3D chat. Avatars in cyber communities should respond to events such as someone entering a virtual chat room or react to objects in their path just as real participants might do (they will either acknowledge the presence of an individual, alter their motion to avoid an obstacle and sometimes fail to notice an event because their attention is otherwise engaged).

While the mapping between motor task and corresponding motion is well understood, the role of attending behavior is less clear and often not specified at all. Also, attention is emergent (where an agent looks changes due to interactions between simultaneous tasks and in response to the dynamics of the environment). Motor actions may in turn be modified by input from the attentional system (e.g., if an agent notices an object bearing down him, he will step out of the way).

Imagine a human character strolling through the park, noticing events, reaching for a paper, waiting for the light to change before crossing a road and generally avoiding objects (small children, pets, toys) that may be in his path. Where should the agent look? What if several events vie simultaneously for the person’s attention? If he stops and just takes in the scene before him, how does the richness and complexity of the environment determine where he looks next?

The AVA takes as input, in text format, a set of actions such as walk to a goal, monitor the traffic light, monitor oncoming traffic. The system generates the corresponding figure animation of motion in Jack and also generates the appropriate eye gaze or looking behavior. The resulting attending behavior reflects interactions or competition between deliberate (endogenous) tasks, involuntary (exogenous) attentional capture and lapses into idling or spontaneous looking. Further, such behavior is generated in real-time.

This method attempts to incorporate and parameterize what is understood about
visual attention from three areas of research: human ergonomics, cognitive psychology and biologically inspired models of computer vision. Essentially, a set of primitive motor activities (walk, reach, lift, manipulate, ...) and cognitive actions (monitor, visually search, visually track...) are associated with predefined patterns of looking behavior. Monitoring activities are additionally associated with memory uncertainty thresholds. Patterns are estimated in this system based on empirical and qualitative data from related experiments in human factors as well as simple observation. In the AVA, looking behaviors implement patterns of eye movements and compete in a psychologically motivated framework. In a multi-task situation or in the presence of exogenous distractors, performance degrades (performance is measured by speed of eye movements to task targets). Interspersed with deliberate looking patterns are lapses into idling.

The inspiration for such a premise comes from experiments done in [Yarbus, 1967] and [Stark and Choi, 1996]. Depending on an observer’s intentions or goals, eye fixations will vary even when directed at the same image. In [Yarbus, 1967], observers were shown a picture and asked to estimate the ages of figures in the picture. Patterns of fixations were directed at the face of each figure. When asked to estimate the “material circumstances” of participants, fixations were directed at the clothes of each figure. Images are scanned in cycles that are composed of repetitive patterns of eye movements. Interestingly, the pattern of fixations remains the same across cycles: the most relevant features for a task are examined in the scene while secondary elements remain unexamined [Yarbus, 1967]. When an agent is engaged in more than one task that requires the same sensory resource, as expected, performance degrades versus the single task condition [Hirst, 1986]. The AVA also encodes involuntary functions known to exist in the human visual system [Hillstrom and Yantis, 1994; Kahneman, 1973; Yantis, 1993]. This method will generate a character’s pattern of attention based on competing voluntary and involuntary behaviors, anticipation, task load, and the environment being viewed.
Related experiments in psychology and vision provide insights regarding specific instances of attending behavior for given circumstances (e.g., visual search, eye targeting for reach motions). The contribution of this dissertation is to provide a framework in which eye movement patterns for specific actions can be combined and interact with each other (providing a simulation of cognitive load) and with exogenous factors (illustrating attentional capture by task unrelated events). This interaction is modeled as a three-level hierarchy in which behaviors related to intentional tasks have the highest precedence, exogenous behaviors the next highest precedence and spontaneous looking (or idling behavior) has the least.

1.1 Background on Eye Movements and Scanpaths Implemented in the AVA Method

The human eye has a visual field of slightly more than 180 degrees horizontal (up to approximately 200 degrees under optimal conditions [Grigsby and Tsou, 1994]) and 135 degrees vertical [Wandell, 1998]. Resolution toward the periphery of this field is an order of magnitude lower than in a high resolution, two degree area covered by the fovea. The fovea is directed toward the object of perception by convergence and divergence of the optical axes and saccadic eye movement (high velocity rotation of the eye). Once the eye fixates on a moving object in situations of visual tracking, however, it follows a pattern of smooth pursuit or slow rotation [Yarbus, 1967].

The eye makes at most 2 fixations per second with each fixation lasting in the range of 250 to 450 msec [Moray, 1993]. Convergence or divergence of the optical axes lasts in the tenths of seconds [Yarbus, 1967]. Durations of fixations undergo minimal changes across tasks. As a task becomes more demanding, however, the frequency of fixations increases [Moray, 1993].

Scanpaths are repetitive cycles of eye movements that an individual performs when looking at a given image [Stark and Choi, 1996; Yarbus, 1967]. Ninety percent of
cycle duration is consumed by eye fixations while only ten percent by actual saccadic eye movement [Stark and Choi, 1996]. When an individual is given an instruction, or behaves with deliberate intent, scanpaths are indicative of the subject’s global problem solving strategy. Scanpaths constrained by a deliberate task tend to be more consistent across subjects [Yarbus, 1967; Stark and Choi, 1996] than in a free viewing scenario. Experiments in [Stark and Choi, 1996] and [Yarbus, 1967] suggest that an internal cognitive map guides task strategy. Scanpaths seem to be a form of foveal sampling that check, verify and validate reality against such a cognitive map. The analog of scanpaths in the AVA are eye behaviors that generate a characteristic pattern and frequency of objects (or locations) that need to be looked at while the agent is engaged in a given activity.

While scanpaths can be viewed as a top-down form of control, attention tends also to be drawn to local regions of conspicuousness in an image, particularly in the absence of deliberate intent [Kahneman, 1973; Stark and Choi, 1996]. This type of behavior is a bottom-up, form of attention capture known in [Kahneman, 1973] as spontaneous looking. Free viewing, or looking in absence of task, is a highly individual and idiosyncratic behavior [Stark and Choi, 1996]. The AVA implements free viewing using a simplified image processing technique (see section 5.6). Such behavior is activated in those instants of time when task demands are not currently active (although the task itself, such as locomotion, may be ongoing).

Deliberate patterns of looking, scanpaths, simultaneously compete in the AVA mimicking the notion of increasing cognitive load. Distractors, known as exogenous events in the psychology literature [Yantis, 1993; Jonides, 1981; Hillstrom and Yantis, 1994], also increase perceptual load by interfering with attention to task related objects when the agent is in a divided or diffuse state of attentiveness (such as visual search). A motion sensing behavior is implemented in the AVA that detects (by querying the scene graph) objects that fall in an agent’s periphery and that have moved inter-frame. While not all such objects may be looked at (overt orienting),
their presence increases response time to deliberate task targets (indicating a covert shift in attention).

A detailed discussion of the psychological inputs to the AVA’s design is presented in Chapter 3, while implemented eye behaviors are examined in Chapters 4 and 5.

1.2 Overview of the AVA Method

The focus of this research is to provide a psychologically plausible framework in which deliberate, involuntary and idling visual attention compete. Also, a set of predefined looking behaviors that correspond to common motor and cognitive primitives are provided in the AVA implementation. Figure 1.1 illustrates the flow of control in the framework.

Users assign tasks to a virtual agent through a menu interface (discussed in section 6.2). Output of a task planner may similarly be used as input to the system. A task queue manager process for an agent consumes and sequences such requests. For each request, the process spawns the appropriate motor activity (such as locomotion) and the corresponding eye behavior. Deliberate eye behaviors that are spawned compete with exogenous attentional capture and lapses into idling. The agent’s attention controller mechanism arbitrates between these three levels of behavior and directs an agent’s line of sight accordingly.

The AVA will animate the relative speed of eye movements (speed is related to which eye behavior is active and the level of perceptual load) and may modify motion based on inputs from the visual system.
1.3 Utilizing the Graphics Database – Assumptions of the Model

The AVA generates attending behavior but does not implement basic vision routines such as object recognition or identification. If an agent is given the task, *search for chair_1*, the goal of the action, *chair_1*, is a named and instantiated figure in the virtual environment (if the figure is not present, a sweep of the visual field is performed). In such an example, the AVA’s visual search eye behavior performs *geometric reasoning* to determine whether the target is visible (see chapter 5.1). If the agent’s line of sight to the target is unoccluded, indicating that the object is visible, then a series of eye movements to the specific target is generated.

The more general command, *search for a red chair*, can be implemented by a specialized version of the AVA’s visual search mechanism. However, “chair” figures in the environment still need to be tagged as such when the graphics database is created. The concept of semantically linking together primitive visual features into a
known object is beyond the scope and not the intent of this research.

The goal of the AVA is to generate eye behaviors in a process that is both psychologically motivated and computationally efficient. To that end, the 3D scene graph (a listing of every object’s edge, vertex, color, position and face information) is utilized by eye behaviors whenever appropriate. Whenever information is known about the agent’s intent, or activity, reasoning through the scene graph rather than vision routines is performed. The only exception to this approach is in the absence of any task information. For example, during spontaneous looking, a simplified image processing method is implemented since free viewing operates at the level of local, visual primitives.

Motion is estimated in the AVA not by optic flow filters but rather by sampling for movement (by querying positions of objects) over an interval of several frames (the shorter the interval, the sooner, more accurately the AVA senses movement). If an object does not fall into an agent’s field of view (this can be determined through geometric reasoning), it will automatically not be considered in the motion sample.

The motivation for querying the graphics database is two fold: a) this research is intended for the virtual reality and computer graphics domain b) a low bandwidth attention algorithm is needed for agent control particularly in applications over the Internet (see following section 1.4 for potential applications of this research).

1.4 Motivation and Applications

While hand-scripted animation can generate realistic and beautifully convincing behavior, it remains unsuitable for applications in dynamic and changing environments and those that require behavior generated at interactive rates. Further, looking is a subtle and emergent indication of intent and cognitive process: one that may not easily be predicted by hand. Disney’s famous Illusion of Life [Thomas and Johnson, 1981], a summary of years of artistic and technical animation experience, provides
guidance on rendering *eye expressions* for various emotions but none on estimating patterns of looking.

Some potential applications of this research are:

- Realistic avatars and participants in cyber-chat communities. A phenomenon is occurring on the Internet: traditional text-based chat rooms are being replaced with graphical worlds where users may converse, interact or merely mill about and observe. The Palace [http://www.thepalace.com/, 1998], created by Electric Communities Inc., is a 2D graphical chat site. Avatars are 2D characters dropped into a flat virtual world. Active Worlds [http://www.activeworlds.com/, 1998], has extended this concept into a 3D distributed, virtual environment using its own 3D geometry standard. Web clients connect to an Active Worlds server and are presented a variety of hyperlinks to various virtual, stylized communities (one may be a city square, another may be a virtual museum inhabited by a company’s products). Avatars are a series of characters that can perform a limited range of key-frame gestures on demand. Field of view may be from an avatar’s perspective or from a panoramic, global camera.

One notices upon “entering” such a community that an odd sort of chaos predominates. While participants can gesture, swivel and alter heading, and talk (by typing text messages in a shared window), it is often difficult to determine who is talking to whom or to get anyone’s attention. Conversational rules of engagement, explored in [Cassell and Vilhjalmsson, 1999], will certainly improve such a scenario. However, realistic avatars should also *look* and respond to events appropriately. Just as might happen in a real gathering, sometimes avatars will fail to notice events because their attention is otherwise engaged. When an avatar walks to a goal, or looks for someone in the community, their behavior should reflect actual eye behaviors (corresponding to locomotion, visual search and response to peripheral events).
• A tool for animators to generate the looking behavior appropriate to a set of tasks an agent should perform. Tasks may be non-manual cognitive activities (like visual search or monitoring) or motor activities like walk and reach. This behavior will also respond to task unrelated events (such as peripheral motion) and the dynamics of the agent’s environment. The set of supported tasks and scope of scenarios for which this research is appropriate is presented in Section 1.6.

• Virtual reality immersive games. Human players anticipate that animated players move and behave appropriately to the circumstances of the game. Since game environments are typically changing, characters’ responses cannot be scripted in advance.

• Determining the ergonomics of computer simulated environments. Performance, in terms of speed of eye movements, is adjusted automatically in the AVA reflecting degradation in ability due to increasing cognitive load or interference from exogenous factors in the environment. This model of eye behavior could indicate when critical events in the environment remain unattended.

1.5 Generating Behavior Versus Generating Plans

This research does not propose a production system approach or unified theory on learning about and dealing with a simulated environment. A system such as SOAR [Lewis, Huffman, John, Laird, Lehman, Newell, Rosenbloom, Simon, and Tessler, 1993], alternately, provides a large, knowledge-based cognitive modeling implementation. Activities input to the AVA need to be entered in a semantically meaningful manner (e.g. the script of tasks needs to make sense) and be composed of supported primitives (e.g. locomotion, visual tracking, visual search, monitoring behavior, etc.).
Rather than an explicit representation of working memory, this research associates patterns and *memory uncertainty thresholds* with activities (uncertainty thresholds determined by empirical data or observation) that determine the frequency with which task related sites are glanced at. Complex rules of engagement are instead encoded in finite state machines that combine visual and motor primitives. Scanpath strategies that combine behaviors such as discussed in 7.1, may be stitched together from primitives implemented in the AVA (for example, walking and visual search). Here, visual and motor routines are patched in such a way that the output of one (e.g., visibility checking in search) is used to regulate the other (e.g., walking along an obstacle free path).

This technique will generate the appropriate *looking behavior* for an agent and can be used to determine if, due to the demands of simultaneous tasks or exogenous factors, critical events remain unattended. The goal of this approach is to automate visual attending behavior in *real-time* with minimal computational overhead.

### 1.6 Scope of Potential Scenarios

The class of scenarios for which this research is appropriate are those which can be generated from combining the technique’s supported set of motor and cognitive primitives. Motor primitives for which this technique supports eye behaviors are: walk, reach, grasp, lift, put down, pull and push. Cognitive primitives in the AVA with associated eye behaviors are: monitor, search (for a target), limit monitor (monitor more frequently under a given circumstance), and visually track. For example, a scenario where the agent searches for a target (e.g. “find the blue table”), walks to the target, reaches for and manipulates an object (e.g. “pick up the newspaper”) and then walks to a destination (“walk to exit”) is a simple case that combines several of this method’s supported primitives. Recall that the AVA technique will generate behavior by considering the *amalgam* of simultaneously executing tasks (since some
tasks such as monitoring and locomotion can proceed in parallel) and by factoring in involuntary looking behavior. In this simple example, if an object flies into the agent’s field of view (and no other task demands are active), the agent will notice and track it. Motion may be modified in the presence of increasing cognitive load (if too many objects are vying simultaneously for attention, motion will slow down). Locomotion itself may be modified by input from visual behaviors (if a collision with a moving object is likely, the agent will stop). Also, an agent will often lapse into idling behavior while a task is active. For example, when walking to a goal, the agent will not need to continuously look either at the goal or the ground in front of him (he will only do so when the memory uncertainty thresholds for those locations are reached).

Additional scanpaths may be added to the AVA if the frequency and general pattern of eye movements for the strategy are known or can be obtained from empirical data. For example, when climbing a ladder, a possible pattern and frequency input to this technique may be to look at the next rung and, when the next step is initiated, look at the following rung. Also, if a particular set of object-specific features are relevant, they may be added to the AVA. When glancing other characters in the simulation, for example, the eyes and mouth of the other agent may be scanned. When tracking a car, the headlights and driver may be looked at.

1.7 Requirements of the Human Model

This research is implemented using the Jack human modeling software. However, any virtual human model which supports a head and eye control mechanism can be integrated with the AVA. Essentially, the AVA method provides either a site (a named 3D location) or 3D location in the environment which the head and eye controller must target. Also, since the method supports eye behaviors for various motor skills, any scenario requiring those skills (e.g., locomotion, reach) will require a human model
that is capable of animating those motor capabilities. Numerous models already exist that are capable of such basic skills (In [Chen, Pieper, Singh, Rosen, and Zeltzer, 1993], the authors propose a system for articulated human figure control. In [Singh, Pieper, Popa, and Guinness, 1993], the authors present techniques for head and eye alignment and facial muscle control of a virtual character).

1.8 Outline of Thesis

This thesis is organized in subsequent chapters as follows:

- I discuss alternate approaches that have been used in generating eye gaze behavior including: robotics and vision research, motion capture, facial animation, conversational agents and image processing techniques.

- I review related work in the psychology literature that provides the basis for the AVA methodology.

- I expand the hierarchy of eye gaze behaviors which compete in this system. I discuss in detail the relationship between cognitive and motor activity with patterns of looking and uncertainty levels.

- I examine a composite behavior combining locomotion with visual search and illustrate how it fits in the AVA framework.

- I provide a worked example illustrating how this technique’s major data structures change and adapt over the course of a simulation.

- I conclude with a discussion of the relative advantages and limitations of the AVA method, some lessons learned over the course of this research and possibilities for extensions to the framework.
Chapter 2

Related Work

This chapter discusses complementary or parallel research involving the determination of visual attending behavior.

Robotics and computer vision researchers are concerned with developing robots that exhibit human like behavior. Also, in computer vision applications, determining the focus of attention aids in reducing complexity of processing (attention acts as a filter that selects which regions of interest to process in camera images) [Brooks, Breazeal, Irie, Kemp, Marjanovic, Scassellati, and Williamson, 1998; Marjanovic, Scassellati, and Williamson, 1996].

Image processing and vision techniques have been developed that attempt to model where humans look in the absence of task. The AVA method incorporates a simplified version of such approaches [Tsotsos, Culhane, Wai, Lai, and Nufflo, 1995; Koch and Ullman, 1985].

Research in animation has explored issues of eye engagement during social interactions or discourse between virtual agents [Cassell, Pelachaud, Badler, Steedman, Achorn, Becket, Douville, Prevost, and Stone, 1994; Cassell and Vilhjalmsson, 1999]. Similarly, visual cues of attention between a robot and a human instructor are explored in [Scassellati, 1996] as are rules of engagement between a virtual tutor and a real student in [Johnson, 1994]. The AVA may be used to extend systems that deal
with issues of facial animation and social interaction of virtual agents.

Motion capture methods, including eye tracking, are used to replay pre-recorded motion or behaviors. While considerably more accurate than the AVA method, such techniques are essentially scripted and unsuitable for dynamic simulations.

2.0.1 Cog

M.I.T.’s Cog Project [Brooks, Breazeal, Irie, Kemp, Marjanovic, Scassellati, and Williamson, 1998; Scassellati, 1996; Marjanovic, Scassellati, and Williamson, 1996] aims to develop a humanoid robot which learns or acquires skills during its interactions with its environment.

Two experiments relating attention and action in this research are visually guided pointing [Marjanovic, Scassellati, and Williamson, 1996] and estimating a human instructor’s line of sight [Scassellati, 1996]. In the pointing task, Cog’s head and eye controller mechanism learns the mapping between locations in the environment being viewed (a camera image) and the appropriate joint angles necessary to align the head and eyes with a target. Also, visual feedback is provided to the robot’s arm control mechanism as a means of learning how to point to the visual target. A motion detection algorithm is used to determine the end point of the arm. Since the predicted position of the arm (within the center of field of view) and actual position may differ, the corresponding error term is used to tune weights in the arm control algorithm. In this experiment, attention is categorized as a neighborhood in the camera image where the arm end point is expected to be. Similarly, interactions between the Cog robot and a human instructor are examined in [Scassellati, 1996]. Such work explores issues such as responding to an instructor’s attentional cues and pointing to request shared attention.

Unlike the AVA technique, this research is concerned with acquiring or learning behavior from the ground up (i.e. as a child might learn how to fixate and point to targets). Issues of competing events and interference from exogenous factors is not
addressed and is not the intent of such work. Also, in the AVA, motor behavior may be modified due to increasing cognitive load. However, unlike Cog, it is understood that the agent has sufficient experience to perform a basic set of motor skills such as walk, reach, grasp, etc.

2.0.2 Conversational Agents, Social Interaction and Intelligent Tutors

Limited rules of eye engagement between animated participants in conversation are discussed in [Cassell, Pelachaud, Badler, Steedman, Achorn, Becket, Prevost, and Stone, 1994] based on psychological observations from [Argyle and Cook, 1976]. Looking behaviors such as head nods to signify turn taking in conversation, using gaze to determine how an utterance is being received and using gaze to accompany accent or emphasis are defined. The domain of this work, similar to research in modeling interactions between a human-like robot and instructor in [Scassellati, 1996], relates rules of social interaction, social cues and feedback with looking behaviors.

Bodychat [Cassell and Vilhjalmsson, 1999] and Thorisson’s [Cassell and Thorisson, in press] emotional feedback work, also address the use of gaze as a significant component of communicative behavior. Expanding the work done in [Cassell, Pelachaud, Badler, Steedman, Achorn, Becket, Douville, Prevost, and Stone, 1994], these projects outline and implement gaze behaviors used in conversation (e.g. sustained gaze to initiate conversation, raised eyebrows and looking away to indicate turn taking, and looking away and lowering eyebrows to plan an utterance).

Rickel and Johnson’s virtual intelligent tutor [Rickel and Johnson, 1997], Steve, has a perception module which is used to monitor changes in the virtual world. The perception mechanism is used to monitor events in an actual student’s field of view and can feedback changes to the tutor’s planning system.

The AVA differs from the preceding projects since its methodology is concerned with the demands of simultaneously executing cognitive and motor tasks as well as
exogenous effects. Rules of social interaction can, however, be imported into the AVA as a composite scanpath behavior which then competes with other ongoing agent activities.

2.0.3 Facial Animation Systems

Facial animation systems [Parke and Waters, 1996; Kalra, Mangili, Magnenat-Thalmann, and Thalmann, 1991; Pearce, Wyvill, Wyvill, and Hill, 1986] relate expression and facial muscle movement to emotion. Eye expression rather than pattern of eye movement is addressed in such applications. A traditional hand animation reference [Thomas and Johnson, 1981] also discusses various known eye expressions (surprise, anger, happiness) but does not provide guidance for estimating patterns of looking. AVA methodology, in contrast, is concerned with the pattern and frequency of eye movements in general settings.

2.0.4 Motion Capture Techniques

Motion capture and facial tracking systems are used to recreate the behavior of a human actor performing specified actions. Recovering line of sight from facial images is processing intensive [Scassellati, 1996; Marjanovic, Scassellati, and Williamson, 1996] while head mounted eye trackers are cumbersome or, at minimum, movement limiting [Crane, 1994]. In fact, technology for real-time body motion capture, whether optical or electromagnetic, virtually precludes the simultaneous capture of eye motions. Hence, when introducing characters to a changing environment as found in interactive multi-user games, pre-recorded behavior is not sufficient to animate the eyes. Human behavior in such systems should be reactive and even proactive: it cannot be scripted in advance.
2.0.5 Image Processing Approaches

Neural net [Tsotsos, Culhane, Wai, Lai, and Nufflo, 1995; Koch and Ullman, 1985] models of attention map task demands into feature primitives such as color, orientation, and luminance. In order to emulate voluntary task-driven control of attention, spatial areas in an image with relevant features are activated (combinations of important features will receive higher activation). Exogenous, or task unrelated stimulation of attention is modeled by activating areas with high local feature contrast. Such approaches are computationally intensive and are usually applied to a given single task [Koch and Ullman, 1985] or in the absence of any task motivation [Tsotsos, Culhane, Wai, Lai, and Nufflo, 1995; Koch and Ullman, 1985].

Since the AVA’s goal is real-time animation, the technique operates at the level of object features or sites whenever feasible. In the absence of any deliberate task or exogenous capture, we model a type of idling behavior known as spontaneous looking. We incorporate a simplified image processing technique, explained in Chapter 5.6, to generate such behavior.

2.0.6 Other Related Techniques

Parallel distributed models in the cognitive science literature [Cohen and Huston, 1994; Cohen, Dunbar, and McCleland, 1990] map task features such as color or words into network units. Such models are applied in the context of a single given task. Activation and network weights determine task response times. Noser, Renault, and Thalmann [Noser, Renault, and Thalmann, 1995] used visual-guided agent locomotion in their work. Approaches in the visual display design literature examine which preattentive visual features should be used and combined to convey information in a manner that requires the least processing overhead [Lohse, Biolsi, Walker, and Rueter, 1994; Healey, Booth, and Enns, 1996]. All these techniques are usually difficult to generalize and, other than Noser’s locomotion work, not applied in the context of combining motor activity and attention.
Terzopoulos’s artificial fish project [Terzopoulos, Tu, and Grzeszczuk, 1994] implements a vision module that determines the identity and location of nearby fish (by querying the graphics database). Feedback from this sensor is used to manage schooling and avoidance behaviors. While this work is related to the AVA in that it links motor behaviors and perception, no notion of a psychologically motivated framework based on human perception or competition between deliberate and idling behaviors is presented.
Chapter 3

Related Psychology Experiments - Basis for the Method

A human character’s attention is directed by volitional, goal-directed aims known as *endogenous factors* that correspond to the current task(s) being performed. Involuntary attentional capture by irrelevant stimuli such as peripheral motion or local feature contrast are said to be *exogenous factors* [Yantis, 1993].

The demands of a particular task generate a characteristic *pattern* of eye movements. Depending on an observer’s intentions or goals, eye fixations will vary even when directed at the same image. Such repetitive cycles of eye movements are known as *scanpaths* [Stark and Choi, 1996; Yarbus, 1967] and indicate the moment by moment task execution strategy employed by an individual. Scanpaths in the AVA are represented as eye behaviors that generate which objects or locations need to be attended by the agent. Monitoring eye behaviors are additionally associated with memory uncertainty thresholds. Such behaviors will generate relevant objects (or locations) that need to be attended at particular intervals (for example, a locomotion eye behavior will generate the horizon and the ground at infrequent intervals) [Moray, 1993].

The transitioning between simultaneous tasks is characterized in [Allport, Styles,
and Hsieh, 1994] as “shifting intentional set.” When engaged in more than one task that requires the same sensory modality, performance degrades versus the single task condition (a review of divided attention experiments is found in [Hirst, 1986]). This phenomenon is realized in the AVA by increasing response time to task targets as the number of events vying for an agent’s attention increases.

Attention may be directed covertly without explicit shifts of gaze or overtly. Covert shifts of attention are measured by line-motion illusion [Hikosaka, Miyauchi, and Shimojo, 1996], brain activity increase in the V5 area using functional MRI [Rees, Frith, and Lavie, 1997] or facilitated response times to targets in attended regions [Posner and Cohen, 1980]. Overt shifts in gaze are preceded by shifts in attention [Klein and Pontefract, 1994]. The AVA seeks to characterize the observable effects of attention shifts relevant to character animation. Hence, covert shifts are relevant in so much as they interfere with or increase response time to targets [Jonides, 1981] in unattended locations.

Once attention has shifted, perception of targets in the attended location is facilitated as long as targets appear within 100ms of the shift [Posner and Cohen, 1980]. If targets appear after 300ms, an increase in target detection time occurs [Posner, Rafal, Choate, and Vaughan, 1985]. This phenomenon is known as inhibition of return and accounts for attention shifting through space.

When attention is not engaged, eye saccades to targets are within the order of 100ms and are known as express saccades [Fisher, 1986]. When a character is attending to a task, however, eye saccade time between relevant sites will increase to 200ms [Fisher, 1986]. Voluntary engagement of attention acts as a “hold mechanism” [Allport, 1993] and suppresses express saccades to irrelevant stimuli. The tendency to orient gaze toward irrelevant distractors is found in patients with frontal-parietal brain lesions [Ladavas, Zeloni, Zaccara, and Gangeni, 1997] (reflecting impairment of oculomotor control) and in early infancy [Johnson, 1994] (reflecting the underdevelopment of selective attention). This range of behavior is characterized in
the AVA method by a distractability parameter that allows a probabilistic sampling of irrelevant stimuli.

What sorts of exogenous factors capture attention and with what frequency? A review of the literature suggests that peripheral events [Jonides, 1981] and abrupt onsets, the introduction of new perceptual objects into a scene, capture attention [Yantis, 1993] when attention is in a diffuse or divided mode (i.e. the target may appear anywhere). However, when attention is fully engaged in a particular location, capture by onset does not occur [Yantis and Jonides, 1990]. Similarly, functional imaging of brain activity indicates that perception of motion, even in the periphery, is reduced or eliminated when attention is fully consumed by current task demands [Rees, Frith, and Lavie, 1997]. Since onsets appear to be rare phenomenon in general settings, the AVA attempts instead to detect and predict interference (to attended objects) from peripheral events.

Moving objects within the center of view do not necessarily capture attention unless motion detection is a necessary feature of the given task [Hillstrom and Yantis, 1994]. Motion generates an onset when it segregates an object from a surrounding perceptual grouping [Hillstrom and Yantis, 1994]. Generally, feature singletons, perceptual features that differ from their backgrounds by color, motion or orientation, interfere with goal directed attention only when the task itself requires “singleton detection mode” [Egeth and Yantis, 1997; Folk, Remington, and Wright, 1994].

In the absence of any given task, attention follows patterns of spontaneous looking [Kahneman, 1973] where areas of high local feature contrast capture interest.

In summary, it is known that tasks impose a voluntary pattern of eye movements. As several tasks are simultaneously attempted, performance degrades. Peripheral events capture attention when the agent is engaged in a task which requires diffuse attentiveness (i.e. visual search or divided attention). In the absence of tasks or peripheral stimuli, attention follows patterns of spontaneous looking.
3.1 Patterns of Looking Associated with Cognitive and Motor Activities

Avionics engineering studies have constructed memory uncertainty models that predict the allocation of attention when monitoring cockpit instruments [Moray, 1993]. The AVA generalizes such activity by incorporating uncertainty thresholds in monitoring eye behaviors. For example, locomotion is treated as a generalized class of monitoring task. Uncertainties may be modified due to increasing cognitive load or changes in state of the underlying monitored object.

Visual tracking also is treated in the AVA as a type of monitoring activity. However, if the target becomes occluded by another object in the environment (and is no longer visible), the eye behavior predicts (based on target heading and velocity), the appropriate reappearance point.

Visual search behavior is implemented in the AVA by having the agent scan the environment until the object of interest is acquired [Rao, Zelinsky, Hayhoe, and Ballard, 1996; Rabbit, 1983]. If the object is not visible, or not present in the environment, a sweep of the visual field is performed. A composite visual search and locomotion behavior is discussed in section 7.1. In such a composite eye behavior, visibility information guides the agent’s walking strategy.

I model eye behavior that corresponds to reach and grasp motions by looking at the relevant grasp sites. Once the hand is in close proximity to the goal site, we initiate attention behavior for the next motor action in the task queue. The notion of when to initiate the next eye movement is examined in Chapter 6.1.

I model two classes of exogenous eye behavior: capture by motion in the periphery of view and spontaneous looking. Spontaneous looking, a term coined in [Kahneman, 1973], is a pattern of “free viewing” eye movements in the absence of any explicit task. The AVA uses a technique motivated from image processing approaches in [Koch and Ullman, 1985; Tsotsos, Culhane, Wai, Lai, and Nufflo, 1995].
Chapter 4

AVA Architecture

The AVA framework attempts to unify what is known about visual attending from psychology, human ergonomics and computer vision into a simple computational framework. Presented at a high level in section 1.2, this chapter expands details of the architecture. Figure 4.1 illustrates the AVA design.

Scanpaths are implemented as eye behaviors for broad categories of motor and cognitive activity. Such behaviors are realized using C++ PatNets, a finite state control language for animation developed at the University of Pennsylvania’s HMS Lab [Noma and Badler, 1997] (finite state control of animation appears also in [Zeltzer, 1982]). PatNets may execute simultaneously and in parallel. Computational aspects of PatNets are examined in section 4.1. Eye behaviors generate a pattern and frequency of eye movements by adding objects and locations to be attended to a list known as Intentionlist. This list identifies sites or objects (corresponding to deliberate actions) that are currently vying for attention.

A peripheral motion sensor determines (using geometric reasoning) those objects in agent’s periphery that are moving. Such objects are added to a list known as Plist. All moving objects will not necessarily be attended (when the agent does look at such an object, the behavior embodied is attentional capture by exogenous, peripheral motion). Appearance changes, such as flashing, are not sensed as motion. Such
Figure 4.1: Method Architecture
changes are a form of abrupt onsets (see Chapter 3). With a minimal computational overhead, the motion sensor behavior (see section 5.5) in the AVA can check for appearance changes (by querying the display status of objects in the graphics database) as well as checking for motion.

A GazeNet, an arbitration mechanism that determines where an agent looks, instant by instant, is illustrated in Figure 4.2. The algorithm used in GazeNet processing is described in Figure 4.3.

A spontaneous looking, or free viewing eye behavior, is activated in those instants when there are no deliberate or exogenous events vying for attention (this behavior is discussed in section 5.6).

Behaviors of the same type compete equally for an agent’s attention. Competition of deliberate behaviors occurs because several parallel tasks may demand attention at different instants. If a group of tasks require attention at the same instant in time, then task objects are added in FIFO order to Intentionlist. The Intentionlist is not implemented as a priority queue because it is unclear what, if any, priorities exist between deliberate visual primitives. The exception to this processing occurs when a moving object is noticed. If the object is on collision course with the agent, it is added to the beginning of Intentionlist (since attending to such an object becomes paramount). Monitoring frequencies are an implicit form of priorities in the AVA. If monitoring an event becomes more urgent (for example, if a particular condition in the simulation becomes true and the agent notices that the condition is true), then the monitored object is added more frequently to Intentionlist (thus causing the agent to look at it more often).

Behaviors of different types compete in a hierarchy. Task related eye behaviors have the highest precedence. As the number of concurrent task eye behaviors increase, response time to targets increases. A probability factor is used to determine overt orienting toward peripheral stimuli (such a factor is age and personality dependent). If the agent is engaged in visual search or in a series of parallel tasks (requiring divided
A: Examine Intention List

!empty(IntentionList) or !empty(PList)

empty

Examine PList

empty

Look at Task Related Site * 

Sample Irrelevant Stimuli?

no with probability \(1 - D\)

yes with probability \(D\)

A:

*: Time to complete look may be function of IntentionList length, Plist length (if in divided attention mode)

Figure 4.2: GazeNet
attention), the presence and number of peripheral events will increase response time to task-related targets. Spontaneous looking has the lowest precedence and can be interrupted by any other type of behavior.

Users enter requests in text format which are stored on a task queue. Sample requests and task queue processing are examined in section 6.2. The Taskq manager net parses requests and spawns the corresponding eye behavior (as well as a corresponding motor action if needed). Eye behaviors add objects, locations, or relative angles (for visual search) to Intentionlist which then compete for an agent’s attention (see Figure 4.1) along with exogenous and idling behavior. If multiple actions are put on the queue, eye behavior that corresponds to a subsequent action may be initiated before a previous motor activity completes (confidence levels and interleaving are discussed in chapter 6.1).

Table 4.1 summarizes the PatNets implemented in the AVA. Each behavior is presented in expanded detail in the following chapters. Some of the nets summarized are representative of broader classes or types of attention.

For example, the motion sensor net samples for peripheral motion and exemplifies exogenous attention capture. Other examples of exogenous attention are capture by feature singletons (abrupt onsets, isolation of color or shape). Since singletons are difficult phenomenon to measure and their ramifications less understood in a 3D world (see chapter 3), only peripheral motion sensing is implemented.

Monitoring eye behaviors are implemented in several levels of detail. General monitoring associates a frequency of looking at a particular event in order to maintain an accurate view of the event in memory. A special type of monitoring, implemented in the AVA, is a limit monitoring net. These nets are associated with a boolean condition (the limit). If the limit condition becomes true over the course of a simulation, the frequency of monitoring increases. Locomotion and visual tracking are also a class of monitoring behavior associated with special processing (see sections 5.2 and 5.4).

A single example of a composite eye behavior, visually guided locomotion, is
implemented in the AVA. Many such behaviors exist in the real world (for example, conversational rules of eye engagement). The purpose of implementing one is to demonstrate that such complex behaviors can function within the AVA framework.

In the following chapters, I delineate the major components of the AVA. Each type of eye behavior is presented and its function in the overall framework explored. The technique for sequencing and interleaving eye and motor behaviors is presented. Then, the link between the AVA and the jack human model is discussed (as well as a presentation of the head and eye control mechanism). Finally, I discuss the strengths and limitations of the AVA method.

4.1 Properties of Finite State Machine Language

C++ PatNets are finite state machines implemented in a pseudo-parallel framework. Nets are not actually spawned as parallel processes. Every active net is added to a list and all nets in the list are evaluated (or advanced) every frame (simulation step). PatNets are composed of nodes (associated with an action to be performed) and transitions (boolean conditions evaluated each frame). Transitions connect nodes in a net. Additionally, PatNets are associated with the following capabilities:

- global and local variables
- message passing (inter-PatNet communication)
- a syntax that allows for sequenced and concurrently active nodes in a net

Any control language that supports the above capabilities, and is integrated with the animation loop, will work just as well in the AVA implementation.
While not empty(\textit{Intentionlist})

\begin{align*}
\text{Object} & \leftarrow \text{Removehead(\textit{Intentionlist})} \\
\text{If not empty(\textit{PList})} \\
\text{\quad look at head of \textit{PList} with probability } D \\
\text{else} \\
\text{\quad If the Object is a figure, look at its center of mass.} \\
\text{\quad If the object is a site, look at its location.} \\
\text{\quad If an angle, look in direction of angle relative to torso.} \\
\end{align*}

If not empty(\textit{PList})

\begin{align*}
\text{Object} & \leftarrow \text{Removehead(\textit{PList})} \\
\text{Look at Object’s center of mass.} \\
\end{align*}

While empty(\textit{PList}) and empty(\textit{Intentionlist})

\begin{align*}
\text{Do spontaneous looking. Take snapshot of field of view. Determine locally conspicuous pixels.} \\
\text{Aim head and eyes at most conspicuous locations in succession.} \\
\end{align*}

Figure 4.3: GazeNet Algorithm
<table>
<thead>
<tr>
<th>Net</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Sensor</td>
<td>If an object in the agent’s periphery view moves from the previous frame to the current and is not already on PList, add the object to PList.</td>
</tr>
<tr>
<td>Monitor Net</td>
<td>When the memory uncertainty threshold is reached for a task object or location, add the object to Intentionlist.</td>
</tr>
<tr>
<td>Reach Eye Behavior</td>
<td>Add the relevant reach/grasp sites to Intentionlist. When the end effector is close to the target, remove site from Intentionlist.</td>
</tr>
<tr>
<td>Visual Search</td>
<td>Determine target visibility, generate intermediate eye movements to target, or perform a sweep. Add to Intentionlist.</td>
</tr>
<tr>
<td>Composite Search</td>
<td>Visually Guided Locomotion. Perform visual search and use visibility information to guide walking.</td>
</tr>
<tr>
<td>Visual Tracking</td>
<td>Add target object with a track frequency to Intentionlist. If target is occluded, predict and add reappearance location.</td>
</tr>
<tr>
<td>Limit Nets</td>
<td>If the limit condition is satisfied, add object or site to Intentionlist.</td>
</tr>
<tr>
<td>Spontaneous Looking Net</td>
<td>Find locally conspicuous pixels in the agent’s field of view. Convert their locations back into pan and tilt coordinates for the agent’s head and eyes.</td>
</tr>
<tr>
<td>Task Queue Manager</td>
<td>Consume requests. Spawn motor or cognitive behavior for an action. Spawn corresponding attention behavior nets. Wait for motor behavior to complete if actions done in sequence.</td>
</tr>
<tr>
<td>GazeNet</td>
<td>Arbitrate between three levels of eye behaviors: intentional, exogenous and spontaneous.</td>
</tr>
</tbody>
</table>

Table 4.1: Eye Behavior Nets
Chapter 5

Eye Behaviors in the AVA

This chapter examines in detail the eye behaviors that make up the AVA framework. Eye behaviors can be classified as deliberate, exogenous or idling. Deliberate eye behaviors implemented in the framework are: visual search, monitoring and locomotion, reaching and grasp and visual tracking. The notion of exogenous attention capture is implemented as a peripheral motion sensor. Visual idling is implemented as a spontaneous looking behavior.

5.1 Visual Search

The first step in visual search is determining whether the target is, in fact, visible. In order to emulate search behavior, if the target is not present in the agent’s field of view, a series of angles is generated that will perform a sweep of the visual field. The scene graph is queried to determine if the target exists in the environment. If it does,
the angle between the agent’s current line of sight and the target is determined:

\[ v_1 = \text{unit vector representing agent line of sight} \]

\[ v_2 = \text{unit vector between center of eyes and target center of mass} \]

\[ \theta = \arccos(v_1 \cdot v_2) \]

Projecting \( v_1 \) and \( v_2 \) onto the x-z plane:

\[ v = v_1 \text{proj} \times v_2 \text{proj} \]

if \( v[1] \geq 0 \)

\[ \text{direction} = \text{counterclockwise} \]

else

\[ \text{direction} = \text{clockwise} \]

(5.1)

The target is in the agent’s field of view if \(|\theta| \leq 90.0\).

If the target falls within the field of view, equation 5.1 is used to find the angle between the agent’s line of sight and every figure in the environment. All targets that lie in the same direction and are approximately the same angular distance from the agent’s line of sight are added to a \textit{Potentials} list. A ray is cast (by calculating the rotation of the line of sight vector \( \theta \) degrees in the target direction) toward the target.

The intersection between this ray and every figure bounding box in \textit{Potentials} is calculated and a record of the closest figure (to the agent) maintained. If the closest figure \textit{is not} the target figure, then the target is not visible (Intersection tests are done in 2D for efficiency. A more accurate test would be intersection of ray and segment bounding boxes – for each segment in a figure). This test can be made more accurate at the expense of some computational overhead. For example, if a small object is directly in front of a much larger target, then this test will return that the target is occluded (it may, however be visible on either side of the occluding figure). Performing a visibility check using immediately \textit{neighboring} rays clockwise and counterclockwise will handle such a situation (e.g., if the target is visible along any of the three three rays - direct, clockwise, or counterclockwise, then it is visible.

33
If the target is not visible, then a sweep of the visual field is performed otherwise a sequence of angles that moves the eye from its current position to the target location is generated. Angles are added to Intentionlist in order. Visual search thus proceeds in a a low to high accuracy manner. If the target is in the visual field, a series of progressively more accurate movements are made to it (rather than immediately acquiring it). This eye behavior corresponds to experiments and a computational model proposed in [Rao, Zelinsky, Hayhoe, and Ballard, 1996]. When asked to locate a specific object in a scene, subjects in [Rao, Zelinsky, Hayhoe, and Ballard, 1996] performed a series of (progressively more accurate) eye saccades toward the object rather than immediately fixating it.

As the agent executes the scanpath generated, the target may become visible (either because the agent or the target moves). If the target falls along the agent’s line of sight during execution of a search scanpath, then remaining scanpath angles are removed from Intentionlist (in essence, aborting any subsequent search).

If the agent is walking, a new heading is chosen along a ray of greatest unoccluded depth (such a ray can be calculated since the relative angles to all figures in the environment are computed as part of the visibility check). Figure 5.1 illustrates reasoning performed in the visual search eye behavior. The ray in red is used to determine whether the target is visible. The rays in mauve are the ones determined to be unoccluded (from calculation in the visibility check) and may be used as a new direction heading. Figure 5.2 shows a simulation of a walk and search behavior where the agent has been assigned the task of finding a candy cane in a forest.

Although the visibility test requires knowledge of target position (and the agent, in fact, does not have this knowledge), the goal of the visual search behavior is to generate an appropriate scanpath relative to an agent’s knowledge of the world. Casting a ray to target position is essentially a form of geometric reasoning using the graphics database. For example, when choosing a new heading, any ray of unoccluded
Figure 5.1: Visibility Checking
depth may be followed (not necessarily one closest to the target). Target position then is not made directly available to an agent unless the target is in fact visible and noticed.

If all rays in an agent’s field of view are unoccluded, as illustrated in Figure 5.3, the composite walk and search behavior will make the agent turn around and continue in a new direction. Figures 5.4 - 5.6 show snapshots from such a simulation.

5.2 Monitoring and Locomotion

Experiments in Moray [1993] examine how often pilots glance at cockpit navigational equipment. The authors cite that pilots verify the state of a given device’s signal at intervals predicted by Nyquist Sampling (twice the signal’s frequency). However, a pilot’s attention often returns to a previously sampled signal earlier than predicted due to interference from increasing cognitive load.

In the AVA, monitoring tasks (locomotion being a general case) use uncertainty thresholds [Moray, 1993] that relate how often a signal, event, or goal should be glanced at in order to maintain an accurate view of the signal’s state in memory. When the uncertainty threshold for a given monitoring task is reached in the AVA system, the relevant site is added to Intentionlist. The time taken to complete eye movements will vary with increasing load and exogenous distractors.

Uncertainties thresholds are useful since they allow ongoing monitoring tasks to be interspersed with other activity. While walking, for example, an agent in the system looks toward the horizon or destination and occasionally glances at the ground [Swain and Stricker, 1993]. The agent does not gaze fixedly at his destination or the ground, but only when memory uncertainty requires that he do so. This is an example of a monitoring task with high uncertainty thresholds. If the state of the terrain changes, becoming slippery or uneven, the uncertainty threshold associated with the ground plane could be reduced, causing the agent to glance more frequently at the ground in
Figure 5.2: Looking for the Candy Cane
agent

Figure 5.3: Change Heading if All Rays Unoccluded

front of his feet.

5.2.1 Limit Monitoring

Monitoring may also be associated with limit conditions [Moray, 1993]. As a signal’s state approaches a critical or cautionary level, it will occasion more frequent eye fixations. For example, when crossing the road, an agent will more frequently glance at the light or crossing signal if it is yellow rather than green.

Obstacles, or objects in an agent’s path, may be considered limit signals in the system. Such objects will not occasion eye fixations until the agent is in very close
proximity.

LimitNets are associated with object sites and a boolean function that indicates when a limit condition has been reached. If such a condition becomes true, the site(s) in the net (i.e. the center of the red light panel in the road crossing example) are added to Intentionlist.

5.3 Reaching and Grasp

Traditional experiments indicate that eye movements precede hand movements and since eye saccades are extremely fast [Abrams, Meyer, and Kornblum, 1990], the eye arrives before the hand motion is started. However, as the target size increases beyond extremely small (0.5 degree visual angle), eye and hand movement are initiated almost concurrently [Bekkering, Adam, van den Aarssen, Kingman, and Whiting, 1995].

When initiating a reach and grasp motion, the AVA generates eye movement toward the relevant grasp site by adding it to Intentionlist. If an agent is picking up a cup, the method generates a glance at the cup handle. If the agent is lifting a box, it generates a sequence of eye motions to the box grips. Clearly, the eye is supposed to establish targeting for the hand [Abrams, Meyer, and Kornblum, 1990]. The AVA animates the reach motion by calling the Jack human model's inverse kinematics routines. The alignment constraint on the eye is established through the algorithm described in Figure B.1.

When the hand is in close proximity to the grasp site, the AVA begins eye movement to the next fixation site (either as a result of the current reach or due to the next motor action on the queue). Associate with motor activity is a confidence or repeat factor. If the agent has done the same reach before or the action is a priori marked with a high confidence factor, abandonment of the current eye behavior happens earlier than it would otherwise. Figure 5.7 shows the agent completing a reach motion and placing a grasped object in a goal location (the box – here the agent
glances at the goal of the “put down” action).

5.4 Visual Tracking

Visual tracking is treated in the AVA as a specialized class of monitoring behavior. A tracking uncertainty is associated with target objects (while currently defined at the beginning of a simulation, this value could be related to target velocity: the faster a target moves, the lower uncertainty and more often sampling should progress). The tracking uncertainty indicates how often the agent should sample the object’s heading in order to maintain an accurate view of the object in memory (at periodic intervals, tracking behavior will add the target object to Intentionlist).

The human eye perceives a moving figure if it appears in the visual field for more than 0.15 seconds and travels at a speed greater than 1 minute/second [Yarbus, 1967]. The eyes are directed toward the moving target by saccadic eye movement. Once the eye fixates on a moving object, however, it follows a pattern of smooth pursuit [Kahneman, 1973].

If a target moves behind an occluding figure, tracking behavior will predict (as humans do) an approximate reappearance location based on target heading and occluding figure size (Currently, reappearance time based on target velocity is not considered, since occlusions are not usually such large figures that a significant time elapse occurs. Tracking behavior can easily be updated, however, to add a predicted location at a given reappearance time). Figure 5.8 illustrates tracking reasoning as a target passes behind an occlusion.

The tracking behavior will add the predicted location to Intentionlist and sample for the target at the reappearance location. In case the target does not appear by the time the agent has looked at the predicted location, tracking behavior will add the last perceived target location to Intentionlist. Tracking eye behavior will continue to alternate between sampling predicted versus last perceived target location until
the target is regained (or the tracking behavior itself is turned off).

Tracking eye behaviors are associated with a duration condition (as are other monitoring behaviors) that specify how long the behavior should remain active (this may be a number of animation frames or or a boolean related to environment conditions).

Figures 5.9 - 5.11 show a simulation where the agent is asked to track the moving ball as well as car objects. In frames 10 and 17, the agent tracks the ball. By frame 36, the agent is tracking the car. At frame 45, the agent attention manager attempts to track the ball once again, but it is no longer visible. A prediction of the ball reappearance point was made (see the green ray). At frame 57, the agent once again tracks the car. Finally, at frame 93, the ball target is regained.

5.5 Motion in the Periphery

Motion in the periphery is a salient event. However, an individual may or may not overtly orient (e.g., make an eye movement) towards such an event. Functional imaging of brain activity indicates that perception of motion is reduced or eliminated when attention is fully consumed by current task demands [Rees, Frith, and Lavie, 1997]. However, until this attentional capacity is reached, perception of such motion can’t be ignored (illustrating involuntary attentional capture).

The presence of such motion increases response time to task targets (indicating covert orienting of attention) when the agent is in a divided or diffuse state of attentiveness (e.g., engaged in a task where the target may appear anywhere) [Yantis and Jonides, 1990]. The AVA models such interference by increasing response time to task targets in the presence of peripheral motion (for all deliberate tasks other than reach since all other types of task require divided attention). When more concrete experimental data from the psychology community becomes available, a “capacity limit” threshold may be used to determine response to such interference (e.g., When
the number of deliberate objects simultaneously vying for attention crosses a capacity limit, attention is fully engaged and peripheral motion ignored [Rees, Frith, and Lavie, 1997].

In the AVA, a motion sensor behavior is active and queries the scene graph to determine if objects that fall within an agent's periphery have moved inter-frame. Rather than use an optic flow algorithm, the AVA employs geometric reasoning to detect motion. Currently, the motion sensor behavior samples for motion every 5 frames. This sampling time is purely for efficiency of rendering the animation. The shorter this interval, the more average computation per frame (thus increasing rendering time) results, but the more accurately motion estimation is performed. The motion sensor net adds peripheral moving objects to Plist.

An agent's attention controller, a Gazenet (see chapter 4) samples from Plist with a pre-determined probability factor (such a factor is age and personality dependent [Johnson, 1994]) if the agent is already engaged in deliberate activity (otherwise, peripheral motion is always noticed). If the agent “samples” or notices a peripheral event, then the Gazenet performs a heading and collision prediction. If the moving object appears likely to collide with the agent (based on heading and velocity), then deliberate tracking of the object is performed (the moving object is added to the beginning of Intentionlist). Figures 5.12 and 5.13 illustrate such a scenario. The agent notices a moving car (frame 0), performs a heading and velocity prediction (frame 4, indicated by the blue and green lines) that indicate a likely collision. The agent stops (frame 20), tracks the car, and then continues along his original heading (frame 77).

5.6 Spontaneous Looking

In those instants in a simulation where neither deliberate task demands or peripheral motion are active, the AVA emulates “free viewing” or spontaneous looking behavior
[Kahneman, 1973]. When an individual looks at an image without deliberate intent, attention is drawn to items that are likely to be informative or significant. Psychologists argue this is caused by a need to reduce uncertainty about our surroundings. Such behavior operates at the level of primitive visual features (such as color, shading, brightness and orientation) since no task constraints are imposed.

Novel or complex items are considered significant. Novelty may be measured by motion, color, isolation, or complexity of shape. Image processing approaches in [Tsotsos, Culhane, Wai, Lai, and Nufflo, 1995; Koch and Ullman, 1985] look for areas in the field of view that are locally conspicuous. Luminance is considered salient in [Tsotsos, Culhane, Wai, Lai, and Nufflo, 1995] while color and orientation are the measure of conspicuousness in [Koch and Ullman, 1985]. Such systems use neural net, biologically inspired, models of vision where feature primitiv es compete to capture attention.

Although a top-down task strategy is not applied, individuals still perform a characteristic sampling when free looking at an image. Experiments in Stark and Choi [1996] indicate that although a free scanpath is idiosyncratic (not consistent across subjects), a specific individual still performs a consistent, repetitive cycle of eye movements.

In Stark and Choi [1996], free viewing scanpaths are generated for a given image using Markov matrices. An artist marks those areas in the image that are judged to be locally salient. A Markov matrix is defined, with transition probabilities (judged by hand) of switching from a given region to the next.

The AVA employs a simplified image processing approach to estimate free looking. Since the goal of this research is unscripted behavior, generated at interactive rates, neither neural net approaches nor a priori marking an image for salience is suitable. The system copies a snapshot of the agent’s field of view (at most 1100 pixels wide by 500 pixels high) into a buffer. Those pixels whose color values are the furthest from their neighbors in RGB space are considered to be the most novel, or locally

43
conspicuous. Figure 5.14 draws rays between the agent’s eyes and those pixels of the image with the highest color contrast (the location that each ray first intersects a figure in the virtual environment indicates a site that will be looked at).

The top fifteen pixels are chosen (it is unclear from available psychology experiments what this limit should be; most experimental results choose eight to nine regions Stark and Choi [1996]). Applying the inverse of the graphics pipeline rendering transformations, a pixel’s screen coordinates are converted into a line (referred to as l in this discussion) in 3D world coordinates. The zbuffer is read and the value converted to actual depth by using a process discussed in [Akeley, 1991]. Since the zbuffer is non-linear (resolution, or accuracy, of depth values is greater closer to the near clipping plane) on SGI platforms, the value stored in the zbuffer must be scaled by resolution of the zbuffer, as well as distance between near and far clipping planes. The computed depth value can be used to compute a hypothetical plane (referred to as \( P \)): one whose distance from the camera (the camera location is between the agent’s eyes for a field of view window) is the depth computed and whose surface normal is the agent’s line of sight. The 3D point that corresponds to a given pixel is computed by intersecting the computed line, \( l \), through the pixel with the computed depth plane, \( \mathbf{P} \).

Figures 5.15 - 5.21 illustrate an agent spontaneously looking over the course of several frames. The top snapshot indicates where line of sight is directed (via the red ray), while the bottom shows the agent’s field of view and selected pixel (highlighted with a mauve cube). Sharp points (like the tops of trees in the example) and edges of objects are favored by the algorithm.

While the agent is stationary, the same 3D points are cycled. However, if the agent is walking, or has shifted glance to follow a deliberate or peripheral target, then the \( \text{AVA} \)’s spontaneous looking behavior recomputes the most conspicuous pixels (since the image in an agent’s field of view will change across frames).

RGB color space is used in the \( \text{AVA} \) because graphics hardware actually stores
and renders RGB color values (thus accessing such values is computationally efficient). Spontaneous looking behavior uses vector distance in RGB space to determine locally conspicuous pixels. Other color spaces, such as CIE-LUV* Travis [1991], are more perceptually uniform. Vector distance of color values in CIE-LUV* is more closely linked to the human visual system’s sensitivity to certain colors (frequencies) of light versus others. Converting RGB values to CIE-LUV* could be done in the AVA implementation. However, such conversion will increase AVA computation time. Also, it is unclear that converting to CIE-LUV* will significantly improve spontaneous looking results (since other visual primitives such as orientation and edges are not considered). I discuss in section 8.3 some thoughts on extending the AVA’s free viewing algorithm.
Figure 5.4: Agent Scanning Environment for the ball – Visibility Rays Unoccluded
Figure 5.5: Agent Changes Heading and Continues Search
Figure 5.6: Agent Searching for and Finding the ball
Figure 5.7: Glancing At Goal of Action
Figure 5.8: Visual Tracking – Prediction Past Occlusions
Figure 5.9: Agent Assigned Ball and Car Tracking Tasks
Figure 5.10: Agent Tracks Car and Attempts to Track Ball
Figure 5.11: Agent Follows Car and Subsequently Regains Ball Target
Figure 5.12: Agent notices moving car and estimates collision likelihood
Figure 5.13: Agent Resumes Walking along Original Heading
Figure 5.14: Spontaneous Looking - Rays Intersect Features with Local Contrast
Figure 5.15: Spontaneous Looking
Figure 5.16: Spontaneous Looking
Figure 5.17: Spontaneous Looking
Figure 5.18: Spontaneous Looking
Figure 5.19: Spontaneous Looking
Figure 5.20: Spontaneous Looking
Figure 5.21: Spontaneous Looking
Chapter 6

Managing the Task Queue and Motor Control

This chapter details how motor action and attending behaviors are sequenced and interleaved. A task queue manager process consumes task requests that are assigned by the user to an agent. Requests are converted into the appropriate eye behavior as well as any underlying motor activity. Also, the mechanisms that control the agent’s head and eye alignment, reach motor skills and locomotion ability are expanded.

6.1 Interleaving and Confidence Levels

The interleaving of an agent’s attention will happen as a natural consequence of competing behaviors in the AVA system. In contrast, given a set of sequential motor activities, the AVA must determine when to abandon the current eye behavior and initiate a subsequent one. A boolean variable is maintained in each net that implements eye behavior based on a reach or locomotion. This variable indicates an expectation that the current activity will complete successfully. Normally, such a variable is set when the hand is in close proximity to the relevant grasp site or the agent is close to his destination. If an agent is confident or expert, however, this
variable will be set earlier in the execution of the reach motion reflecting greater confidence in the agent’s skill. Setting this boolean variable thus allows attention to be directed to the next activity while the motor system completes the motor task. Notice that if this variable is set at the beginning of the task, the interpretation is consistent with human behavior: it means the agent knows where to reach or walk even without looking at the object or goal.

6.2 Task Queue Manager and Examples

A task queue net consumes action requests posed for the agent and invokes the appropriate eye behaviors. Figures 6.1(a)-(e) illustrate the structure and content of sample requests.

Task requests may be added to the AVA through a menu interface. The user can assign tasks to an agent interactively or by linking the output of a task planner to the system’s input. Menu commands in the Jack environment can be invoked directly in a script known as a JCL, Jack Command Language, file (task planner output may be redirected to such a file and converted to JCL format). Commands available to the user are:

- Walk to a goal.
- Reach for an object.
- Perform Visual Search.
- Perform Visual Tracking.
- Monitor an object.
- Perform a composite walk and search behavior.

Monitoring and visual tracking behavior are associated with a duration condition (when the condition becomes true, the behavior stops). The duration condition is
identified as the name of a C++ boolean function. Monitoring is also associated with a limit condition, also identified as a C++ boolean function. When the limit condition becomes true, monitoring frequency increases (see section 5.2.1).

If 6.1(a) is entered on the queue, the task manager spawns a monitoring locomotion eye behavior, and initiates the walk motor activity for the human model. The monitoring behavior in the system associates an uncertainty threshold of 100 with the goal and 200 with the ground (implying that every 100 frames the agent should glance at the destination and every 200 frames, he should glance at the ground).

If 6.1(c) is entered on the queue, the task manager spawns a reach eye behavior (with the relevant sites on the box passed as arguments) and invokes the human model’s reach and grasp mechanism. The reach eye behavior indicates, by polling end effector position, when the motor activity is close to completion. This value, when set to true, allows the task manager to initiate eye behavior for the next task on the queue.

If 6.1(d) is entered on the queue, a visual search behavior is be spawned. This behavior generates a sequence of eye movements to the target. These intermediate positions are added to Intentionlist.

Figure 6.2 illustrates processing performed by the task queue manager.

6.3 Head-Eye Movement and Motor Control

The AVA can be integrated with any virtual human model that supports a mechanism for head and eye movement control. Additionally, we associate attentional behavior for locomotion, arm reach and hand grasps as well as cognitive actions such as visual search and monitoring.

The animated human model used in the AVA method’s implementation employs inverse kinematics to create arm motion and collision detection between fingers and grasp object segments to animate hand grasps.
The *Jack* human model employs eyes with two degrees of freedom: vertical rotation corresponding to eye tilt and horizontal rotation corresponding to eye pan. Head motion is controlled by a three degree of freedom joint (which controls head tilt, pan and roll). Currently, we set only the head’s pan and tilt parameters.

Target object coordinates are converted into joint angles that manipulate the human model’s head, neck and eyes. The mechanism which controls the model’s head and eye movement is based on a study of eye-head coordination in [Sparks, 1989]. Small gaze shifts produce only eye movement while larger shifts (between 20 to 90 degrees) generate combined head and eye movement. We plan to expand in future work the mechanism that distributes motion between head and eyes. Experiments in [Freedman and Sparks, 1997] indicate that head contribution increases linearly with shift amplitude. The current algorithm which controls the model’s eye-head coordination is given in Appendix B.

Given a final position and orientation, the *Jack* human model’s locomotion system generates the corresponding walking motion. Attention behaviors are therefore appropriately and reactively generated during whatever collision-free path the agent actually chooses during locomotion. And, of course, the perception and detection of obstacles and imminent collisions may themselves modify the path taken. Other locomotion models that use sensed information, such as Reynolds’ flocks and herds, use sensing to control path [Reynolds, 1987]. What we add are the observations that (1) sensing is a resource to be allocated and directed and that (2) sensing takes time.
| (a) | Agent: bill | Action: walkto | Object: | Goal: lamp post | Sites: | Avoid Figures: | Limit Condition: |
| (b) | Agent: bill | Action: monitor | Object: traffic light | Goal: | Sites: red lamp, yellow lamp, green lamp | Avoid Figures: | Limit Condition: function pointer: returns true when traffic light is yellow |
| (c) | Agent: bill | Action: reach | Object: box1 | Goal: | Sites: box1’s left handle, box1’s right handle | Avoid Figures: | Limit Condition: |
| (d) | Agent: monica | Action: search | Object: ice cream truck | Goal: | Sites: | Avoid Figures: | Limit Condition: |
| (e) | Agent: bill | Action: reach | Object: box1 | Goal: table | Sites: table center | Avoid Figures: | Limit Condition: |

Figure 6.1: Sample Action Requests Processed by Task Queue Manager
1: Generate Looking Behavior For Cognitive Task, Proceed Without Wait

2: Generate Looking Behavior and Motion for Motor Task, Proceed Without Wait

Figure 6.2: Task Queue Manager
Chapter 7

Worked Examples

7.1 Visually Guided Locomotion

A composite behavior in the AVA that combines visual search with locomotion is a walk and search process. Such a behavior scans the environment for a target and uses visibility information obtained in the search to guide an agent’s locomotion. A ray with the greatest unoccluded depth is followed. If all rays in an agent’s view are unoccluded, then the behavior causes the agent to turn and change heading (see chapter 5.1).

Figures 7.1 - 7.3 illustrate a scenario where the agent is looking for a candy cane in a field of pine trees. The agent performs visual search, but also responds to peripheral motion. In frame 2, the agent has started the search behavior. By frame 12 and 18, the agent is overtly tracking a peripheral moving object. At frame 40 and 45, the agent has found the target and is lapsing into visual idling. During the presence of peripheral distractors (the moving balls in Figures 7.1 - 7.2), the time taken for the agent to complete eye movements to deliberate task locations is greater than otherwise (indicating covert orienting of attention and thus interference from exogenous effects). In chapter 8, I suggest further work that may be done in modeling interference effects and implications for the resulting eye behaviors.
Frame 2

Frame 12

Figure 7.1: Visual Search and Interference
Figure 7.2: Overt Orienting Peripheral Object and Visual Idling
Figure 7.3: Visual Idling
7.2 Crossing the Road Example

Consider a scenario where an agent is asked to walk to a destination: in order to reach the destination, he must cross a road, watch out for oncoming traffic and monitor the appropriate traffic signal. We ask the AVA to handle such a scenario with the input illustrated in Figure 7.4.

A task queue manager net for agent “Stanley” will consume these actions requests. A walking eye behavior net will be spawned that periodically adds sites to Intentionlist (the sites will be the destination and, infrequently, the ground in front of Stanley’s feet). Also, the walking motor activity will be spawned (the corresponding eye behavior will remain active as long as the motor activity is not complete).

A monitoring eye behavior will be spawned that periodically adds the traffic light as a figure to be monitored on Intentionlist. If the light turns yellow, the frequency with which the monitoring behavior adds the traffic light to Intentionlist will increase. The monitoring behavior will only remain active while the agent is crossing the street.

A monitoring eye behavior will also be spawned to check for oncoming traffic on the right side of the road. If a car approaches within a certain distance of Stanley (5 meters), the frequency of monitoring will increase. This behavior will also remain active until Stanley crosses the street.

Behaviors modeling exogenous factors (involuntary attention capture by task unrelated events) will be a peripheral motion sensor and spontaneous looking. A peripheral motion sensor will sample for moving objects in agent Stanley’s field of view. If a moving object is detected, it will be added to PList.

Whenever both Intentionlist and PList are empty, spontaneous looking behavior will be activated for Stanley. This behavior will determine the pixel in Stanley’s field of view with the greatest local feature contrast, convert the pixel location back into the corresponding 3D environment coordinates and cause Stanley to glance at the appropriate target.
7.2.1 Details of Simulation

Figures 7.5 – 7.12 are some snapshots from the animated version of this scenario. While the complete animation is approximately 360 frames, we examine some representative frames that illustrate how **Intentionlist** (the queue of task related figures or sites that need to be attended) and **PList** (the queue of moving objects in the agent’s periphery) are modified and adapt over the course of the simulation. Also, we indicate when spontaneous looking behavior becomes active. A line is drawn indicating Stanley’s line of sight when looking at task related or exogenous targets.

By frame 6, a monitoring eye behavior has placed the site “traffic light, yellow” on **Intentionlist** (indicating that the agent should look at the traffic light). Also, Stanley’s peripheral motion sensor has noticed that a car object is moving and placed it on **PList**. Figure 7.5(a) shows the close up view of Stanley looking at the traffic light and Figure 7.5(b) shows a small window into Stanley’s field of view.

By frame 17, a monitoring eye behavior has placed the site “road, right” on **Intentionlist** indicating that Stanley needs to look at the right side of the road (to ascertain if other cars are coming). Also, the walking eye behavior has added the figure “table” to **Intentionlist** indicating that Stanley needs to glance at his destination. Figure 7.6 shows Stanley looking right and Figure 7.7 shows Stanley subsequently tracking the car.

By frame 75 (figure 7.8), Stanley looks back at his destination (the table). At frame 96, the traffic light monitoring behavior places a site on **Intentionlist** again (figure 7.9) (indicating that Stanley needs to look at the light to ascertain its color).

By frame 127, the road monitoring behavior has added a site to **Intentionlist** indicating that Stanley needs to look toward the road again (figure 7.10).

By frame 145, both **Intentionlist** and **PList** are empty, so Stanley lapses into spontaneous looking behavior (Figure 7.11).

By frame 211, a moving ball has arrived in Stanley’s periphery (his peripheral motion sensor has placed the ball on **Plist**). Figures 7.12(a) and 7.12(b) illustrate
Stanley tracking the ball.

7.2.2 Analysis of Simulation

There are several important points that should be noted from the animation generated from this scenario.

First, certain predefined data is associated with the tasks entered into the system (the monitoring frequencies, or memory uncertainty thresholds, associated with walking activity and with watching the traffic light and oncoming traffic). However, the AVA generates eye movements that consider the combination of simultaneously executing tasks (and hence produce *timings* of eye movements that differ from the standard uncertainty thresholds). The state of the *Intentionlist* at frame 17, for example, shows two sites that need to be attended and a car object on PList that needs to be tracked. Essentially, the AVA is generating behavior as a result of increasing cognitive *load*.

Second, the AVA generates eye movements that are the result of changes in the environment and not explicitly the result of a task on the task queue. Whenever tasks demands do not require attention, for example, the agent lapses into idling behavior. Also, when a ball flies into the agent’s field of view (see frames 211 and 226), the agent tracks it in the absence of other task demands. Also, when the traffic light turns yellow, the frequency of monitoring increases so that the agent will glance at the light more often than previously.

The plausibility of the AVA method is determined not so much by how it accurately reproduces the empirical data on which it is based but rather in how it *adapts* and in how it *fails*. If too many deliberate tasks had vied for the agent’s attention in this simulation, he would have ignored a potentially critical event (a car bearing down on him!) and been run over. If several moving objects appeared in his field of view as he was walking to the table, one of them may not have been observed and he could have bumped into it (just as would happen during a real life walk in the park).
<table>
<thead>
<tr>
<th>(a)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent:</td>
<td>stanley</td>
</tr>
<tr>
<td>Action:</td>
<td>walk</td>
</tr>
<tr>
<td>Object:</td>
<td></td>
</tr>
<tr>
<td>Goal:</td>
<td>table</td>
</tr>
<tr>
<td>Sites:</td>
<td></td>
</tr>
<tr>
<td>Avoid Figures:</td>
<td></td>
</tr>
<tr>
<td>Limit Condition:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent:</td>
<td>stanley</td>
</tr>
<tr>
<td>Action:</td>
<td>monitor</td>
</tr>
<tr>
<td>Object:</td>
<td>traffic light</td>
</tr>
<tr>
<td>Goal:</td>
<td></td>
</tr>
<tr>
<td>Sites:</td>
<td>yellow lamp</td>
</tr>
<tr>
<td>Avoid Figures:</td>
<td></td>
</tr>
<tr>
<td>Limit Condition:</td>
<td>function pointer: returns true when traffic light is yellow</td>
</tr>
<tr>
<td>Duration Condition:</td>
<td>function pointer: returns true while agent is crossing road</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(c)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent:</td>
<td>stanley</td>
</tr>
<tr>
<td>Action:</td>
<td>monitor</td>
</tr>
<tr>
<td>Object:</td>
<td>road</td>
</tr>
<tr>
<td>Goal:</td>
<td></td>
</tr>
<tr>
<td>Sites:</td>
<td>right side of road (direction from which cars appear)</td>
</tr>
<tr>
<td>Avoid Figures:</td>
<td></td>
</tr>
<tr>
<td>Limit Condition:</td>
<td>function pointer: returns true when car is in close proximity</td>
</tr>
<tr>
<td>Duration Condition:</td>
<td>function pointer: returns true while agent is crossing road</td>
</tr>
</tbody>
</table>

Figure 7.4: Example Action Requests on Taskq
IntentionList: traffic_light.yellow
Plist: car
Spontaneous Looking Active?: No

Figure 7.5: Frame 6 - Stanley monitors the traffic light
IntentionList: road.right, table
Plist: car
Spontaneous Looking Active?: No

Figure 7.6: Frame 17 – Stanley glances right
IntentionList: car, table
Plist:
Spontaneous Looking Active?: No

Figure 7.7: Frame 38 – Stanley tracks the car
IntentionList: table
Plist:
Spontaneous Looking Active?: No

Figure 7.8: Frame 75 – Stanley glances at his destination
IntentionList: traffic_light.yellow
Plist:
Spontaneous Looking Active?: No

Figure 7.9: Frame 96 – Stanley glances back at traffic light
IntentionList: road.right
Plist:
Spontaneous Looking Active?: No

Figure 7.10: Frame 127 – Stanley glances back at road
IntentionList:
Plist:
Spontaneous Looking Active?: Yes

Figure 7.11: Frame 145 – Stanley spontaneously looking
Figure 7.12: Frame 211-226 – Stanley tracking a moving ball
Chapter 8

Conclusions

Believable virtual actors need to exhibit the appropriate attending behaviors in order to be suitably convincing and human-like. Gaze is a significant and often subtle indication of intent and cognitive process. Automating the generation of looking behaviors is an important endeavor since such behaviors are emergent and often cannot be predicted by a manual animation process. Further, synthetic actors in dynamic virtual environments must respond to changing circumstances and exogenous events. Scripted behavior is inadequate in such scenarios.

This thesis proposes a computational framework for generating attending behavior, referred to as the AVA, using empirical and qualitative observations from the psychology, human factors and computer vision literature. This method drives simulated attention in novel, unscripted, and resource-bound ways for a supported set of cognitive and motor primitives. The goal of this research is to generate more natural looking animated human characters by directing line of sight appropriately.

8.1 Contribution

The contribution of the AVA method is a unified, psychologically-motivated framework that generates a character’s visual attention at interactive rates for a given
set of primitives. Deliberate behaviors, the analogs of scanpaths in the psychology literature [Yarbus, 1967; Stark and Choi, 1996] compete with involuntary attention capture [Yantis, 1993; Jonides, 1981; Hillstrom and Yantis, 1994] and lapses into idling or free viewing [Kahneman, 1973; Stark and Choi, 1996]. When information about a task is known, the scene graph is queried for efficiency purposes. When an agent lapses into free viewing or idling, no task constraints are active so a simplified image processing technique is employed. Monitoring tasks (such as locomotion, tracking) are associated with memory uncertainty thresholds (a concept coined in the study of the ergonomics of avionics cockpits). Uncertainty thresholds allow the interleaving of simultaneously executing tasks and idling (e.g., although a task such as locomotion is ongoing, attending to task sites continuously is not required). Figure 8.1 and 8.2 show which types of behavior are active over the course of a simulation (the scenario animated combines visual search, locomotion and the presence of peripheral moving objects similar to the one presented in section 7.1).

Insights provided by having implementing this framework are:

- A defined set of parameters that need to be modeled which impact the observable effects of attending behavior. Such parameters are: probability of overtly orienting toward peripheral targets, an attentional capacity beyond which exogenous targets are ignored, memory uncertainty for monitoring tasks, and limit of spontaneous regions picked (before resuming a cycle).

- A defined vocabulary of looking behaviors for certain motor and cognitive activity: visual search, visual tracking, monitoring and locomotion, limit monitoring, eye behavior for reach and grasp, exogenous behavior (sensing peripheral motion) and spontaneous looking.

- The necessity of separating eye behavior from an underlying motor activity. In the AVA, eye behaviors are implemented separately from an underlying motor action such as reach or walk. This allows for anticipation and interleaving of
Figure 8.1: Competition Between Types of Eye Behaviors
Figure 8.2: Competition Between Types of Eye Behaviors
eye behavior for sequential motor actions (e.g., looking at the target of a reach when a preceding walk action is near completion).

- A defined hierarchy of three levels of eye behavior (endogenous, exogenous and idling) and a proposed method of how these types of behaviors interact. For example, slowing eye movements to task targets in the presence of exogenous distractors (modeling covert shifts of attention) or lapsing into idling.

- A technique of modifying motor activity based on visual inputs (e.g., stop walking if a noticed peripheral object appears on collision course, using the results of visual search to guide locomotion and slowing motion in the presence of increasing cognitive load).

8.2 Limitations

A parameter that is proposed within the AVA framework but not set is attentional capacity. The attentional capacity after which exogenous events fail to register is known to exist [Rees, Frith, and Lavie, 1997] although is not explicitly quantified. When more related experiments in the psychology domain become available, such a parameter may be validly set.

Monitoring uncertainties are set a priori in the AVA since they are domain dependent. For example, monitoring a red light may have more significance in a cockpit simulation versus a stroll in park scenario. Such uncertainties may be adjusted by animating the activity in isolation (e.g., when establishing locomotion thresholds, one can animate the agent walking to a goal and adjust frequencies of glancing at the horizon and ground). Alternately, such data may be known in advance (e.g., the bandwidth of a signal in a cockpit, or the frequency with which a a traffic light changes).

Within a particular simulation, certain instances or classes of peripheral motion may be anticipated or habitual and thus should not be considered as a distractor. For
example, if two agents are walking together, engaged in conversation, the motion of each participant is anticipated. While other types of moving objects may act as a exogenous distractors, the relative motions of each agent should not.

Composite behaviors such as walking and locomotion require some effort to implement. Patching together primitives supported in the AVA often requires shared data structures (e.g., the visibility information obtained from search needs to be analyzed for walking). Developing a composite behavior as a single PatNet (although visual search is still spawned as a primitive) is necessary in order to avoid timing lags during animation.

The AVA suffers from discontinuities in frame rate in certain scenarios. Spontaneous looking, for example, can be a computationally expensive procedure if field of view changes every frame (since a snapshot of an agent’s field of view needs to be taken and the top most locally conspicuous pixels computed). In a situation where the agent is simply walking to a goal, and performing no other deliberate tasks, then AVA computation will be significant. The agent will tend to mostly perform visual idling. A new snapshot of field of view will need to be computed each frame since the view will continuously change due to the agent’s locomotion.

8.3 Extensions

The AVA uses color difference in RGB space to determine a conspicuous location for free viewing. Orientation of edges, brightness, contours and other visual primitives (shading) [Lohse, Biolsi, Walker, and Rueter, 1994; Koch and Ullman, 1985; Tsotsos, Culhane, Wai, Lai, and Nufflo, 1995] are all salient but cannot be considered in real-time. Extending the AVA free viewing algorithm to include regions of conspicuous color may be a first step toward a more inclusive model.

Values such as overall scene luminance impact field of view [Grigsby and Tsou, 1994] and can influence time taken to respond to visual search targets [Wandell,
1998]. The impact of scene luminance can be easily encoded into the AVA. When spontaneous looking is performed, a snapshot of the pixel buffer is taken. The RGB values for each pixel in an agent’s field of view is available to the algorithm. If the average luminance value is computed (luminance can be considered a sum, or weighted sum, of RGB components), this value can be passed to the foveation mechanism in an agent’s GazeNet. If luminance is below a threshold, then time taken to complete eye movements may be increased. Also, the magnitude of field of view used in sensing peripheral motion and in visibility checking can be modified with changing luminance.

8.4 Metrics for Evaluating The AVA

Evaluating the naturalness and accuracy of animations generated from the AVA is challenging since human behavior is both complex and diverse. In future work, there are two alternate approaches that may be employed in judging the effectiveness of animations produced: a qualitative technique that measures how well expectations about a character’s behavior are met by my system and a more empirical approach that compares actual eye movement patterns and timings.

Judging the similarities of eye movement patterns, and determining whether such patterns are motivated by the same cognitive process, is in itself an active area of research [Hacisalihzade, Stark, and Allen, 1992; Stark and Choi, 1996].

8.4.1 Generated Behavior and Viewer Beliefs

A qualitative judgment may be made regarding how well the AVA generates behavior that corresponds to a viewer’s expectations. In a well defined scenario, as in the crossing the road example presented in section 7.2, a viewer will have set beliefs on where an agent should look and at approximately what timings. For example, the agent should scan the right side of the road for cars (or both sides if the road is two way!), glance at the traffic light and occasionally look to his destination. In the AVA,
the scenario is animated by adding a locomotion task, a monitor the traffic light task and a monitor the road task to the agent’s task queue.

To judge whether the AVA generates expected behavior, an animation should be created by failing to spawn a given eye behavior (either monitoring the traffic light, monitoring the road, or locomotion eye behavior) for the scenario. A viewer may then be shown the animation and asked to estimate what is missing (without outlining possible alternatives). This process may be repeated for all given scenario eye behaviors. If the user estimates the missing piece correctly (e.g., the agent never looked at the traffic light, never watched out for traffic or looked where he was ultimately walking) then the AVA has accurately animated an appropriate set “cover” of eye behaviors for the scenario.

This technique will work if the simulation is well defined and the inference of missing pieces relatively clear. If the simulation involves an agent idling and glancing at his surroundings with no clear purpose then, obviously, determining a missing eye behavior is impossible. Since idling behavior is also very individualistic [Stark and Choi, 1996], estimating the naturalness of visual idling is difficult.

8.4.2 An Empirical Approach

One of the goals of the AVA is to generate emergent looking behavior that may not be easily predicted by traditional methods (manual animation). Asking viewers to determine the naturalness of such animations is difficult precisely because combined looking behaviors are hard to predict.

A measure of the AVA’s success will be how it generates looking behavior for combined tasks. If data regarding an agent’s eye movements for a single task is known, then using the AVA to generate looking behavior for combined instances of the same task and validating the results (against actual human performance) will be a useful evaluation of the method. Visual idling will need to be factored out of the experiment (possibly by making the combined task so intense that the agent has little
or no time to lapse into idling behavior) since such behavior is so idiosyncratic.

Consider a scenario where a particular agent’s “sampling” rate of a signal is known (either from actual eye movement data or if the frequency of the signal itself is known) and the agent’s tendency to orient toward a single peripheral motion target quantified (from empirical data). For example, the agent is in a cockpit and glances at (samples) a particular display with a known frequency. In order to create an evaluation scenario, introduce other signals of the same type to the AVA simulation (by adding additional monitoring tasks to the agent’s task queue). Also, introduce peripheral motion to the simulation (by animating moving targets in the agent’s field of view).

A comparison then needs to be made between the pattern and frequency of eye movements generated by the AVA and the actual human’s performance in a similar scenario (the same person from whom single task eye movement data is obtained should be used to determine performance results). The AVA simulation may need to be run many times to get an “average” of resulting eye behavior. Similarly, human performance should be measured many times to get an “average” indication of performance.

Since visual idling is not considered (and spontaneous looking turned off in the AVA during the animation), time lags between actual and animated performance should not be significant. Actual human performance is the baseline case against which the AVA and any other model of visual attention should be compared. The closer predicted data “fits” actual data, obviously, the more successful the model.

Comparing eye movement data (or scanpaths) for similarity is in itself a research topic. In [Hacisalihzade, Stark, and Allen, 1992], eye movement scanpaths are considered the result of a Markov process. Regions of interest in an image are mapped to a fixed number of states (regions picked by an artist for salience). The probability of transitioning from one state to another is predetermined (based on observed data). Scanpaths are considered “strings” where a region in the image is assigned a letter of the alphabet. Similarity of strings, or scanpaths, is measured using a string editing
technique (substituting, deleting or inserting letters – each operation associated with a given cost). The similarity of strings is measured in units of string edit distance. The authors find that strings generated from the same Markov process, with a small number of fixed states, and few transition probabilities, will fall within certain distance thresholds of each other. This technique is sensitive to the initial choice of regions selected and also to the assignment of string edit cost. While this metric may be informative, the underlying assumptions need to be validated further (e.g., that a scanpath can be modeled accurately as the outcome of a Markov process – particularly when multiple eye behaviors compete in a single simulation).

8.4.3 Summary of Metrics

I discuss two techniques for validating effectiveness of AVA generated behavior. The first is a qualitative measure that compares whether a viewer’s expectations of behavior are met by the AVA method. Animations are generated with particular components missing. A viewer is then asked to identify which type of behavior is lacking (e.g., the agent should glance at the traffic light while crossing the road). A second, more quantitative approach, may be to compare a particular individual’s performance with actual eye movements predicted by the AVA. In a constrained scenario with little idling eye behavior, human eye movements may be measured to determine frequency of monitoring a given signal. This information can be input to the AVA as a monitoring uncertainty frequency. Subsequently, assigning two signal monitoring tasks to the simulated agent, one can compare the frequency and pattern of looking generated by the AVA with actual individual performance. Validating the similarity of eye movement patterns is a challenging task and is, in itself, an active area of research.
Appendix A

Modification of Data Structures

The Intentionlist is a list data structure that stores objects, locations or relative angles (angles used in visual search), related to deliberate looking behavior, that are currently vying for an agent’s attention. The list is modified as follows:

- Eye behaviors add objects to the end of Intentionlist. If a moving target is noticed from PList, and on collision path with the agent, it is added to the beginning of the deliberate list (since monitoring the moving object becomes very important).

- Data is consumed (by an agent’s GazeNet) from the beginning of Intentionlist.

- If the target of visual search becomes visible during execution of the search (either because the agent or the target moves during the simulation), then all remaining angles that are part of the search are removed from Intentionlist. In essence, the visual search strategy is aborted (because the target becomes visible). Angles are identified with a unique identifier, the address (a C++ pointer) of the visual search net that added them to the list. Such angles may appear at any position in Intentionlist.
The **PList** is a list data structure that stores objects in an agent’s periphery of view that are moving. Essentially, the list is a queue except that objects that have aged in information value can be removed from any position in the list. A motion sensor eye behavior samples the environment (every five frames) and adds objects that have moved between samples (and are in the agent’s peripheral field of view) to the end of **PList**. Objects are consumed from the beginning of the list by an agent’s **Gazenet** (indicating attentional capture by a moving object). The motion sensor eye behavior also removes objects that are on **PList** but are no longer in the agent’s field of view. Essentially, such objects are not noticed in time and are no longer relevant (these obsolete objects may be at any position in **PList**).
Appendix B

Head Eye Coordination Algorithm

This algorithm is executed every frame while eye tracking for a given target is active. Hence, head and eyes can be continuously aligned with a moving target. The pan and tilt displacement needed to align the head with a target is calculated independently of the displacements necessary to align the eyes. Horizontal gaze shifts greater than 20 degrees or vertical shifts greater than 10 degrees produce combined head and eye movement. In the combined motion scenario, the eyes move as far as oculomotor limits (joint limits set in the human model) allow. Once the head is aligned with the target, correct realignment of the eyes to the target occurs in the next frame.

In the human visual system, the vestibulo-ocular reflex (VOR) stabilizes images on the retina by generating compensatory eye motion of equal magnitude and in the opposite direction to head motion. For large gaze shifts, VOR activity is turned off until line of sight is aligned with the target. Then, VOR resumes and eyes may counter rotate to compensate for head motion. [Sparks, 1989]

The mechanism of eye head coordination for the Jack human model was implemented by [Xiao and Achorn, 1994] and is given in Figure B.1.
For each eye,
determine view vector between center of eyeball and target
transform this vector into the eyeball coordinate frame
solve for:
\[ \theta_{\text{left or right},v} = \text{vertical displacement} \]
and
\[ \theta_{\text{left or right},h} = \text{horizontal displacement} \]
needed to move eye coordinate frame to the target

Similarly, for the head,
determine view vector between base of head and target
transform this vector into the base of head coordinate frame
solve for:
\[ \beta_v = \text{vertical displacement} \]
and
\[ \beta_h = \text{horizontal displacement} \]
needed to move the head coordinate frame to the target.

If \((\theta_{\text{left},v} > 10 \text{ degrees}) \) or \((\theta_{\text{right},v} > 10 \text{ degrees}) \) or
\((\theta_{\text{left},h} > 20 \text{ degrees}) \) or \((\theta_{\text{right},h} > 20 \text{ degrees}) \) then
\[ \theta_{\text{left},v} = \theta_{\text{left},v} + \beta_v \]
and
\[ \theta_{\text{right},v} = \theta_{\text{right},v} + \beta_v \]
\[ \theta_{\text{left},h} = \theta_{\text{left},h} + \beta_h \]
and
\[ \theta_{\text{right},h} = \theta_{\text{right},h} + \beta_h \]
threshold \( \theta_{i,j} \) so that displacement is within
joint limits
Else
\[ \beta_v = 0 \] and \[ \beta_h = 0 \]
(only the eyes move for small shifts)

Let:
new head position =
\((\text{current head pan} + \beta_h, \text{current head tilt} + \beta_v)\)
left eye position =
\((\text{current l. eye pan} + \theta_{\text{left},h}, \text{current l. eye tilt} + \theta_{\text{left},v})\)
right eye position =
\((\text{current r. eye pan} + \theta_{\text{right},h}, \text{current r. eye tilt} + \theta_{\text{right},v})\)

Figure B.1: Head-Eye Coordination Algorithm
Bibliography


