

Decision Networks for Integrating the Behaviors of Virtual Agents and Avatars

Thomas S. Trias, Sonu Chopra, Barry D. Reich, Michael B. Moore,
Norman I. Badler, Bonnie L. Webber, Christopher W. Geib
University of Pennsylvania
Center for Human Modeling and Simulation
Philadelphia, PA 19104-6389
trias@gradient.cis.upenn.edu

Abstract

We examine the use of decision networks in animating virtual agents. We have developed a system that allows the realization of multiple, parallel behaviors for an agent. The networks we utilize, called PaT-Nets, are used both to represent individual behaviors and also to encode rules of engagement between agents. The multiple networks simultaneously attached to an individual agent are used to control locomotion, planning, visual attention and decision-making strategy. We discuss how human players may be substituted for autonomous players and still operate under the represented behaviors in the PaT-Nets.

1. Introduction

There is considerable recent interest in virtual agents and avatars that populate simulated worlds [8, 15, 18, 21, 22]. A virtual agent reacts to and makes changes in the virtual world it inhabits. We are primarily interested in human-like agents [5, 22]. An avatar is a virtual agent controlled by a human participant: the participant provides both the decision-making and motion behaviors, while the avatar mimics or maps these movements into animations [2]. The virtual agent, or avatar, is bound frequently by rules of the world such as maintaining contact with the current ground plane, avoiding passage through walls, or satisfying “physiological” needs such as sustenance, health, or mere survival. More interesting behaviors arise from mutual interactions of virtual agents and avatars [8, 13], especially when, as with humans, there are social rules to follow, spatial goals to achieve, tasks to accomplish, and roles to play [3]. While various games permit a user to be immersed in a world populated by other creatures and even other players, most of these are either of the search and destroy variety, or else only allow the user the protagonist’s role, to which the other agents simply react. If we are to allow virtual environments

populated by avatars and virtual agents, we must be able to represent their interactions and the “rules of engagement” in a semantically consistent, easily visualizable, and computationally powerful fashion.

We examine the use of decision networks in animating virtual agents. The decision networks, known as PaT-Nets [7, 9, 10], are used both to represent individual behaviors and also to encode rules of engagement between agents. Attached to a particular agent, multiple networks may simultaneously control locomotion, planning, visual attention, and decision-making strategy. Networks may also be agent-independent and simply control aspects of the simulation.

PaT-Nets are finite state machines with message passing and semaphore capabilities. Nodes are associated with processes that can invoke executable behaviors, other PaT-Nets, or specialized planners. An arc transition between nodes in a PaT-Net may check a local condition evaluated within the PaT-Net or a global condition evaluated in an external environment.

We discuss our implementation of a virtual, immersive game in the *Jack*[®] software environment. Section 2 describes the particulars of a game of virtual hide and seek [17]. Then, we examine the locomotion, visual attention and strategy components of our game and how PaT-Nets are used in implementing each subcomponent. Next, we summarize why our networks are particularly useful in virtual reality simulations. Finally, we discuss capabilities and constraints introduced in the game with real players.

2. Virtual Reality with PaT-Nets

To illustrate the flexibility of our approach, we implement a game of virtual hide and seek in our simulation system known as ZAROFF. The simulation can be completely autonomous with virtual hiders and a virtual seeker. Alternately, *either* role can be played by a real participant.

Usually virtual games cede all strategic decisions to the human participant. The virtual agents in such games then *react* to the real participant's actions and behavior. Our game allows a human participant to respond to strategic decisions made by a virtual agent. Unlike other virtual games, the role of primary protagonist, the seeker, can be generated entirely by our simulation.

We discuss first the particulars of our game. We then examine subcomponents which implement decision making (planning), visual attention and locomotion. Note that planning, as well as the other subcomponents, are driven by a controlling PaT-Net, which we call a PLAYNET.

2.1. Rules of the Game

In the ZAROFF system, one may augment or replace the game's default high level controller, which is encoded as a PaT-Net. This allows direct control of the rules and strategies employed within the game. This is done by creating a state machine, via PaT-Nets, with actions that take place at the nodes and transitions that occur when certain conditions are met. More in depth discussions of the PaT-Net schema occur in [7, 9, 10].

Here we describe the high-level controller, or PLAYNET. Complex actions (such as "run home" or "look for hiders") are encoded into subroutines, encapsulating the actions and removing the implementation details from the user of the system. These actions act as a conduit between the PLAYNET, the highest level of the ZAROFF system hierarchy, and the lower level behavioral simulations (such as locomotion and visual attention). The actions must be cleanly preemptable, allowing the simulation programmer to pass in a condition for premature termination, which allows an agent to effectively "change its mind" when certain events occur. Certain complex actions invoke a high-level planner called ITPLANS [11, 12]. The planner's function is to expand them into a contextually appropriate structure of more basic actions that can then be carried out, imbuing the agents with a certain level of deliberative intelligence. By altering the logic behind the scheduling of these high-level actions, one can dramatically alter the manner in which the game is played, providing the agents with different behavioral patterns and constraints. Thus, with minor manipulation to the control structure, one can restructure the simulation to fit different goals or to experiment with alternative behaviors.

Figures 1 and 2 represent two possible games of hide-and-seek. The synchronization state causes players to wait until all the players have reached home and the seeker may begin counting. Hiding involves selecting a location based upon fitness criteria and then instantiating a locomotion controller to effect the transit. In addition, re-selection occurs if a hider notices that another is approaching his desired hiding place. After the culmination of the hiding step, the hider

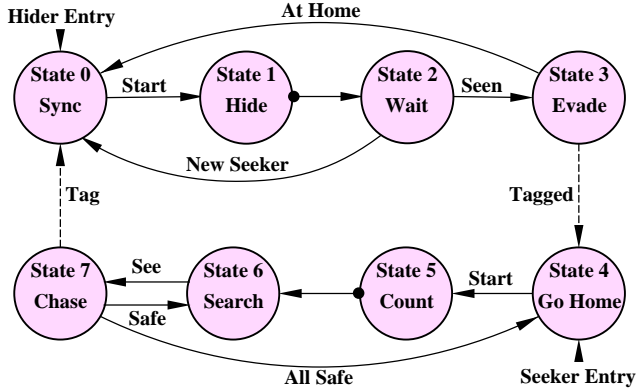


Figure 1. Traditional Rules

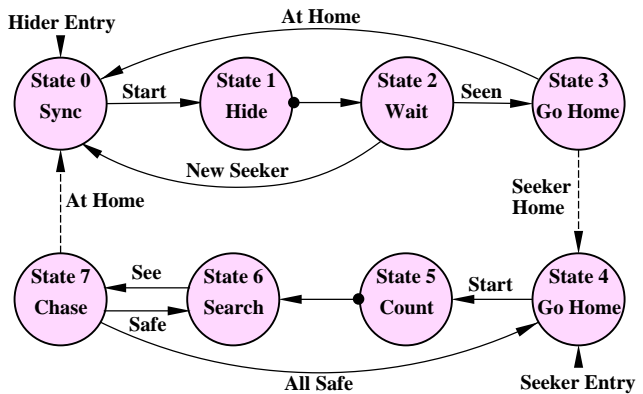


Figure 2. Foot-Race Rules

will wait in his/her hiding place until seen by the seeker. Evasion consists of a primary goal of avoiding the seeker and a secondary goal of getting back to home base. Simultaneously to the hiders actions, the seeker counts (with eyes averted/closed) and then begins searching the environment for hiders. When a hider is encountered, the seeker performs the appropriate action (depending upon the variant of the game). If the hider successfully reaches a “safe” state, then the seeker is forced to resume searching.

The dashed transitions represent a change in role from hider to seeker or *vice-versa*. The transitions with black circles represent the natural culmination of the process described in the node from which they originate. The “at home” condition evaluates to true when the agent is at home-base. The “safe” condition evaluates to true when the seeker notices that the hider has made it home. The “new seeker” condition indicates that a new player has been made seeker and that the game is restarting.

Figure 1 represents a traditional version of the game, in which hiders, once hidden, may not move until seen by the seeker, at which time the hider attempts to run home without being tagged by the seeker. Currently, the seeker indicates having found the hider with an explicit message. Figure 2, on the other hand, represents a “foot-race” variation of the game; upon seeing a hider, the seeker and the hider race home, with the hider becoming the next seeker if the current seeker gets there first. In both cases, the conditions for transit are prioritized to avoid conflicts. Note that each player has a copy of the PLAYNET, which stores state information that is relevant to that particular player. Actions occur in a time-sliced simulated-parallel fashion.

The complex action of searching is realized through an iterative, reactive, hierarchical planner which in turn relies on a special purpose search planner to instantiate unbound references, such as “Hider.” The translation from the PLAYNET to actions is either direct (when the choice of action at a PLAYNET node is obvious) or mediated through the planner (when deliberation is required to choose the action to take). An example of this is when the seeker takes the action Look-For Hider, where a combination of the hierarchical planner and a special purpose search planner combine to generate a sequence of actions designed to locate a hider.

2.2. General purpose planning

The planner, ITPLANS, is a hierarchical planner, in which hierarchical expansion only takes place to the degree necessary to determine the next action to be carried out. It consists of an incremental expansion of the frontier of the plan structure to successively lower levels of abstraction. The incremental nature of the plan allows the system to make commitments at the appropriate level of detail for action while not committing the system to future actions that

might be obviated by changes in the world. The close coupling of ITPLANS with the environment manifests itself in two ways:

First, the traversal and pruning process the planner follows at each interval relies on being able to determine the actual state of the world and to compare that with its goals. During the expansion process ITPLANS examines the state of the world and its memory to determine if any of the goals within its plan have been satisfied. When a goal has been satisfied serendipitously, it can be pruned out of the plan structure, and the system can move on to consider its next goal.

Second, ITPLANS “leans on the world” [1] when predicting the results of its actions. Rather than maintaining a complete model of the world and the state that results from executing the action, ITPLANS uses a simpler method based on associating conditional add and delete lists with each action. ITPLANS assumes that a given proposition is true in the state that results from the action if (1) the proposition is explicitly added by the add list or (2) the proposition is true *now* in the world and it is not contained on the delete list. By this method, ITPLANS can make predictions about the results of executing an action without modeling the entire world state.

2.3. Search planning

Agents, like the humans they simulate, have a limited field-of-view (Section 2.4). A consequence of this limited perception is the need to find objects when they are somewhere outside the agents’ field-of-view. Our approach is to isolate this reasoning in a specialized module, a *search planner*, that translates information acquisition goals to high-level physical goals to explore parts of the environment [16]. To search for an object, an agent must know (or discover during the search) the regions of space where the object might be.

Searches terminate successfully when a referent object is “seen” in the environment. They terminate unsuccessfully when there are no more regions to explore. A search may also be terminated if the environment changes in a way that obviates the search. For example, when the seeker notices that all the hiders have successfully returned to base, there is no point in continuing to search unexplored parts of the environment.

Searches are conducted by using locomotion to move the agent around the environment and then a *visual search* (Section 2.4) is performed. This combination of movement and visual scan ensures that an agent has an opportunity during the course of a search to see every part of space in the game environment.

Our approach to search planning relies on maintaining information about the state of a heuristic search on an in-

ternal map. The heuristic search has as its goal finding a desired object. It uses distance from the agent to order regions for exploration. Two lists of regions are maintained by the search algorithm, an *open* list of regions yet to be explored and a *closed* list of regions which have been explored. Its internal map consists of nodes that correspond to bounded regions connected by links that correspond to doors.

In one iteration of the search, the closest region on the open list is selected to be explored. ITPLANS generates a plan for going to and exploring that region, opening any doors necessary along the way. After executing each action in this plan, ITPLANS observes the resulting world to determine if the desired object has been located. New doors and regions observed during the action are added to both the map and the open list.

Pemberton and Korf [19] present optimal algorithms for heuristic search on graph spaces, where only a portion of the graph is available before the agent must commit to an action. We use their Incremental Best-First Search (IBFS) algorithm, which uses best-first search to find the closest known open node. Heuristic estimates for this known part of the graph are recalculated as necessary.

2.4. Visual Attention

Since an agent's field of view is restricted and his attention limited, he must actively deploy attention to maximize the useful information available to him from the world. In human agents, evolution has also led to subconscious mechanisms for controlling attention to reduce the load on another limited resource, conscious mental processes. For our work, we develop a network that controls an agent's visual attention. Note that this network runs in *parallel* with other networks that control an agent's locomotion and actions.

To be believable an agent must direct his gaze naturally. To an observer or other participant in an activity, random or uncontrolled gaze is both misleading and disconcerting. If a human participates in the virtual game, the gaze behavior of simulated opponents will be interpreted as reflecting their decision-making or cognitive processes. If a human merely observes two simulated agents, the players' gaze behaviors again determine the validity of the simulation.

Gaze also constrains which parts of the environment are visible to a player. For example, hide and seek is driven by the seeker's limited perception. Since a seeker only has knowledge of areas that are immediately visible to him, he must explore his surroundings to discover a hider (Section 2.3).

Various gaze behaviors are attached to hidere and seekers. A seeker performs visual search to investigate the environment. Once he has determined the best location to begin a search, he will fixate upon such a location. As the seeker

moves about in his environment, his attention will be divided between monitoring the surroundings for a target and using gaze as an aid in navigation and locomotion. Hiders will scan the environment in cycles for potential hiding places. As the demands of a task increase, *frequency* of fixations increase. So, as the seeker's count down approaches its limit, a hider will perform more frequent fixations.

Note that one gaze behavior may be preempted by another. An agent's gaze may be captured by unexpected or novel items – something flying into his field of view. Also, gaze reflects decision making and the incremental nature of problem solving. A seeker's gaze may have been directed to a particularly promising location, but if a hider appears in the periphery, he will change direction and fixate on the latest position in which the hider appeared.

Figure 3 illustrates the high-level PaT-Net that implements gaze behavior. The four behaviors which are scheduled and interleaved are attract, avoid, visual search and spontaneous looking (visual pursuit). In the case of the seeker, he initially performs a visual sweep, or search, of his environment. If the agent is attracted to a particular point (the place to commence search for a hider), he will interleave gaze between the fixation point and occasionally the ground. If something flies into an agent's field of view, all other gaze behavior is preempted and the agent's gaze will follow the new object since visual pursuit is involuntary. Scheduling is done by uncertainty thresholds. If an agent has been walking to an attract point for some time, focusing on that point, his uncertainty about the environment increases. When his level of uncertainty reaches a threshold, the agent will perform visual search again. Avoid gaze behaviors are linked to obstacles in an agent's path. An agent generally glances at such obstacles only when they are in immediate proximity. The search, pursuit, attract and avoid gaze behaviors are implemented as their own PaT-Nets.

A hierarchy of PaT-Nets are thus used to schedule behaviors. A high level GAZENET acts as a controller and enforces an attentional template. Additional, lower level nets are used to activate specific gaze behaviors such as visual search. Also, gaze behavior is implemented in *parallel* with other behaviors such as locomotion and action. Unlike other state-table driven animations where parallel behavior is difficult or impossible to implement, PaT-Nets allow multiple, simultaneous behaviors.

2.5. Action Execution and Locomotion

The Action Execution module is responsible for the control of all actions occurring in ZAROFF. Most actions, such as opening and closing doors, Action Execution performs directly. Human locomotion is a special case which is performed by the Behavioral Simulation System (BSS) [5, 6]. Action Execution controls this locomotion indirectly.

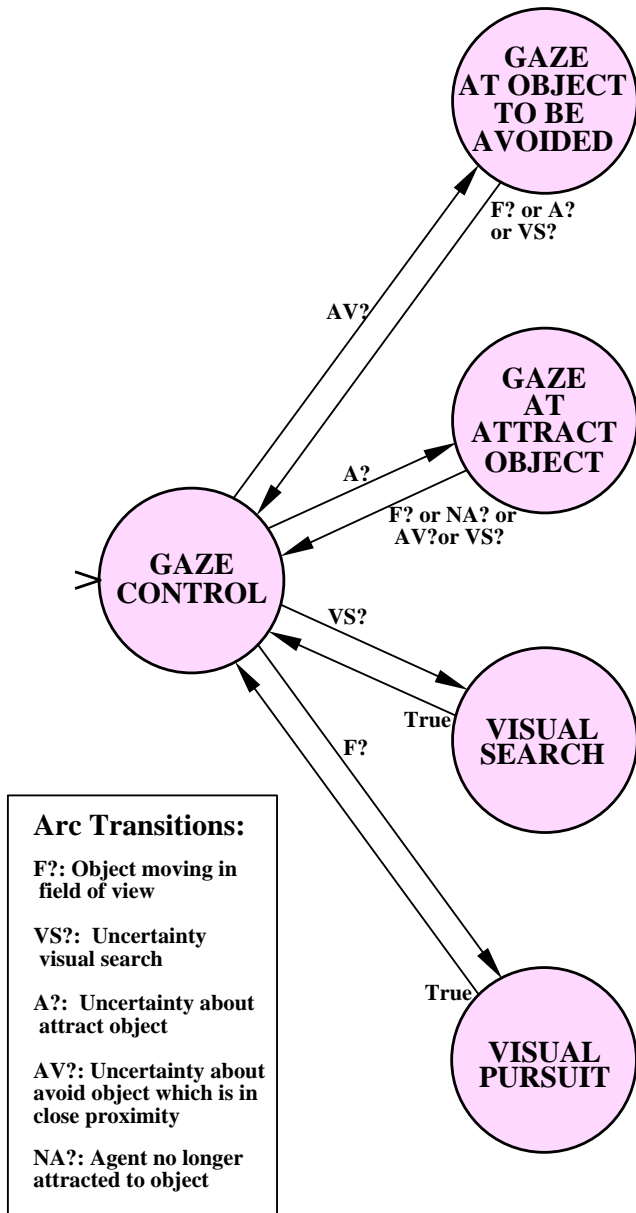


Figure 3. Gazenet: Visual Attention Controller

BSS provides general locomotion of objects in *Jack*, and is used in ZAROFF to generate human locomotion. The central control mechanism of BSS is a loop that includes perception, control, and action. During the perception phase agents sense their environment using a set of simulated sensors. During the control phase the next foot position is selected, and during the action phase the step is taken.

2.5.1. Non-Loconotion Actions. Non-loconotion actions are performed directly by Action Execution manipulating the environment. For example, a door is opened by rotating it about its hinges. This rotation is done incrementally, a small amount each frame of animation.

2.5.2. Human Locomotion. Neither path-planning nor explicit instructions drive agent locomotion; agent control and apparent behavioral complexity result from the interaction of a few simple “behaviors” with a complex and changing environment. A behavior is a function mapping an agent’s state in the environment to the stress of being in that state. It affects the manner in which the agent behaves under specified circumstances. An agent learns about its environment through the use of a network of simulated sensors. Based on the information gathered by these sensors, the path through the terrain is computed incrementally allowing the agent to react to unexpected events such as moving obstacles, other agents, a moving goal, or the effects of limited perception [14, 20].

3. Extension with Real Players

Replacing automated with agents by avatars can be done in a variety of ways. Except for the constraints necessary to maintain the plausibility of the simulation (like gravity and non-interpenetration of objects), avatars may have greater or lesser degrees of freedom. However, the players are still subject to role-induced constraints, which are often difficult to enforce. If an avatar is granted too much freedom, then the human player may be able to force uninteresting or incorrect behavior on the part of the virtual agents by refusing to comply with the rules of the game. However, too little freedom makes the virtual experience uninteresting for the human participant.

The automated players select from among alternative choices during the game using a variety of strategies. For the most constrained interaction, the human controlling an avatar may be presented with a menu of the current choices and asked to select one. Currently, the planner chooses among a set of regions for the seeker to explore and selects the most promising one. A real player could be given the same choice of regions and could manually indicate which one to explore. Necessary behaviors would be strictly enforced.

Alternately, one could allow the human complete freedom of action and movement, thus making the simulation a truly immersive experience. Exception handling routines may be utilized to recognize and handle improper behavior, though it is much better design to avoid these situations in the first place. For example, if the avatar is the seeker, the screen must go blank while the count goes to 10 since there is no other way to force the seeker to “close his eyes.” Also, due to the novelty and excitement of virtual reality, participants initially tend to stand near motionless, ogling at the scenery. Incentive must be provided to force the player to hide, explore, and give chase to other agents. Although, it would not be disastrous for a player to just wander the environment, it would not provoke any truly interesting behavior on the part of the automated players. The rich repertoire of behaviors ascribed to the automatons in ZAROFF are what make it something more than the standard human/virtual environment interaction.

4. Conclusion

The ZAROFF system allows the user to encode different sets of rules and strategies in the intuitive structure of PaT-Nets, and is the first step towards transforming rules encoded in a more natural language into executable code. By imposing a limited, well-defined semantics upon PaT-Nets, it should be possible to encode the rules in a format such that strategy can be automatically generated from the set of rules, as opposed to the current method whereby the user specifies both the rules and associated behavior. With the advent of an upcoming graphical interface to PaT-Nets construction of tailored simulations becomes even easier, especially as the repertoire of encoded actions is expanded.

PaT-Nets lend themselves readily to the control and scheduling needed for animation or virtual simulations [3, 4]. Networks may be used for high level synchronization such as interleaving action and decision-making. Various behaviors such as gaze and locomotion are controlled by parallel, independent networks.

Augmenting the system to incorporate direct human participation would require little or no alteration of the present system, with some added modules to accommodate constraints and input/output. The reactive and decisive nature of the automated agents allows easy expansion of the environment, as the computer-controlled agents treat a human agent no differently from each other.

Typical virtual environments are populated with agents that are solely reactive. The human participating in such virtual experiences is considered to be the focal character and all other agents in the environment take their cues from him. In ZAROFF the human need not fulfill the role of protagonist. The system can generate agents who play the focal role.

5. Acknowledgments

This research is partially supported by DMSO DAAH04-94-G-0402; ARPA DAMD17-94-J-4486; U.S. Air Force DEPTH through Hughes Missile Systems F33615-91-C-0001; Air Force DAAH04-95-1-0151; NSF IRI95-04372; ARO DURIP DAAH04-95-1-0023; and ARPA AASERT DAAH04-94-G-0362. We wish to thank Brett Douville and Jeffry Nimeroff for comments and contributions to editing, and Roxana Cantarovici for her work on hiding-place selection. We would especially like to thank Welton “Tripp” Becket for his work and advice on PaT-Nets.

References

- [1] P. Agre. The dynamic structure of everyday life. Technical Report 1085, MIT Artificial Intelligence Laboratory, 1988.
- [2] N. Badler, M. Hollick, and J. Granieri. Real-time control of a virtual human using minimal sensors. *Presence*, 2(1):82–86, 1993.
- [3] N. Badler, B. Webber, W. Becket, C. Geib, M. Moore, C. Pelachaud, B. Reich, and M. Stone. Planning for animation. In N. Magnenat-Thalmann and D. Thalmann, editors, *Interactive Computer Animation*. Prentice-Hall, 1995.
- [4] N. Badler, B. Webber, W. Becket, C. Geib, M. Moore, C. Pelachaud, B. Reich, and M. Stone. Planning and parallel transition networks: Animation’s new frontiers. In S. Y. Shin and T. L. Kunii, editors, *Computer Graphics and Applications: Proc. Pacific Graphics '95*, pages 101–117. World Scientific Publishing, River Edge, NJ, 1995.
- [5] N. I. Badler, C. B. Phillips, and B. L. Webber. *Simulating Humans: Computer Graphics Animation and Control*, chapter 5, pages 137–150. Oxford University Press, 1993.
- [6] W. Becket and N. I. Badler. Integrated behavioral agent architecture. In *Proceedings of the Third Conference on Computer Generated Forces and Behavior Representation*, pages 57–68, Orlando, Florida, March 1993.
- [7] W. M. Becket. The Jack Lisp API. Technical Report MS-CIS-94-01, University of Pennsylvania, Philadelphia, PA, 1994.
- [8] B. M. Blumberg and T. A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proc. ACM SIGGRAPH '95*, pages 47–54, 1995.
- [9] B. Douville. PaT-Net user’s guide. Technical report, Department of Computer and Information Science, University of Pennsylvania, 1995.
- [10] B. Douville. A theoretical description of PaT-Nets. Technical report, Department of Computer and Information Science, University of Pennsylvania, 1995.
- [11] C. Geib. Intentions in means-end planning. Technical Report MS-CIS-92-73, Department of Computer and Information Science, University of Pennsylvania, 1992.
- [12] C. W. Geib. *The Intentional Planning System: ItPlanS*. PhD thesis, University of Pennsylvania, 1995.
- [13] J. Granieri and N. Badler. Simulating humans in vr. In R. Earnshaw, H. Jones, and J. Vince, editors, *Virtual Reality and its Applications*. Academic Press, London, UK, 1995.

- [14] H. Ko, B. D. Reich, W. Becket, and N. I. Badler. Terrain navigation skills and reasoning. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, pages 219–227, Orlando, Florida, May 4-6 1994.
- [15] M. R. Macedonia, M. J. Zyda, D. R. Pratt, D. P. Brutzman, and P. T. Barham. Exploiting reality with multicast groups. *IEEE CG&A*, 15(5):38–45, September 1995.
- [16] M. B. Moore. Search plans (dissertation proposal). Technical Report MS-CIS-93-56/LINCLAB 250/IRCS-93-29, Department of Computer and Information Science, University of Pennsylvania, 1993.
- [17] M. B. Moore, C. Geib, and B. D. Reich. Planning and terrain reasoning. In *AAAI Spring Symposium on Integrated Planning Applications*, 1995.
- [18] H. Noser, O. Renault, D. Thalmann, and N. Magnenat-Thalmann. Navigation for digital actors based on synthetic vision, memory, and learning. *Computers and Graphics*, 19(1):7–19, 1995.
- [19] J. C. Pemberton and R. E. Korf. Incremental path planning on graphs with cycles. In *Proc. First International Conference on AI Planning Systems*, pages 179–188, 1992.
- [20] B. D. Reich, H. Ko, W. Becket, and N. I. Badler. Terrain reasoning for human locomotion. In *Proceedings of Computer Animation '94*, pages 996–1005, Geneva, Switzerland, May 1994. IEEE Computer Society Press.
- [21] S. Stansfield. A distributed virtual reality simulation system for situational training. *Presence*, 3(4):360–366, Fall 1994.
- [22] S. Stansfield, D. Shawver, D. Rogers, and R. Hightower. Mission visualization for planning and training. *IEEE CG&A*, 15(5):12–14, September 1995.