Appears in the First International Joint Conference on Autonomous Agents and Multiagent Systems, Bologna, Italy 2002

Anticipating Where to Look: Predicting the Movements of Mobile Agents in Complex Terrain

Randall W. Hill, Jr., Youngjun Kim , Jonathan Gratch
Institute for Creative Technologies
University of Southern California
13274 Fiji Way, Marina del Rey, CA 90292-7008
+1-310-574-5700
{hill, yjkim, gratch}@ict.usc.edu

ABSTRACT
This paper describes a method for making short-term predictions about the movement of mobile agents in complex terrain. Virtual humans need this ability in order to shift their visual attention between dynamic objects—predicting where an object will be located a few seconds in the future facilitates the visual reacquisition of the target object. Our method takes into account environmental cues in making predictions and it also indicates how long the prediction is valid, which varies depending on the context. We implemented this prediction technique in a virtual pilot that flies a helicopter in a synthetic environment.

Categories and Subject Descriptors
I.2.10 **Vision and scene understanding**. *Perceptual reasoning, representations, data structures, and transformation*.

General Terms
Algorithms, Performance, Design.

Keywords
Perception, Prediction, Mobile Agent.

1. INTRODUCTION
In this paper we present a method for making short-term predictions about the direction that a mobile agent will travel as it traverses complex terrain. This capability has been integrated with the perceptual system of a virtual human who pilots a helicopter in a synthetic world. The motivation for developing a method for anticipating where to look is rooted in the need for visual attention in virtual pilots. For a number of tasks, the pilot's visual attention is divided between tracking one or more vehicles and scanning the environment for information. To accomplish this involves shifting the pilot's gaze from the vehicle(s) being tracked to other objects in the environment. Figure 1 illustrates a pilot shifting attention between two tanks on a curved road. When the tanks are far enough apart, it can be difficult to reacquire a moving object after a number of seconds have passed, since their paths are not simple linear projections and they can travel at high speeds along these roads. Looking away even for a few seconds can result in losing track of a vehicle since it can move hundreds of feet in a short time.

To make it easier to visually reacquire a moving vehicle, we wanted to enable the pilot to make short-term predictions of where the vehicle will be located up to seven seconds in the future. With this prediction the pilot would be able to shift back his gaze to approximately the right place to reacquire the target object. But projecting the direction and location of a vehicle is not a simple matter—terrain features such as rivers and mountains, and cultural features such as roads and bridges, can strongly influence the path taken by a driver. We do not believe it is sufficient to predict a vehicle's location by making a simple linear projection. A driver may choose to turn at a road intersection or change direction to avoid a natural obstacle such as a lake or a steep mountain. Moving at 48 kilometers/hour, a vehicle can cover 100 meters in the short time the pilot glances away, or worse, the vehicle may change direction and end up someplace unexpected. In either case, the observer needs to reacquire the visual target with a minimal amount of search.

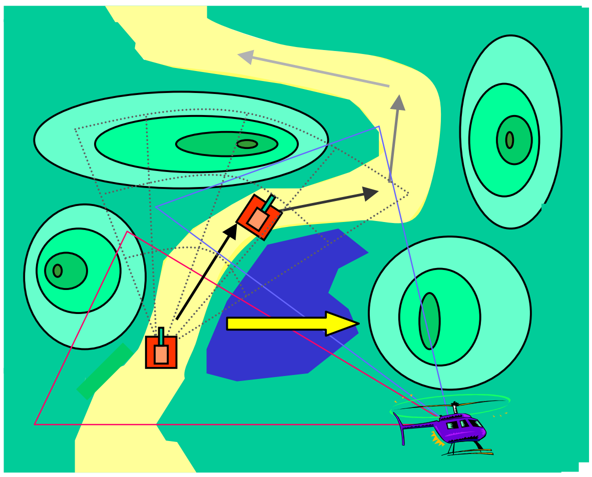

Figure 1: Tracking and reacquiring moving entities

We hypothesize that the environment provides visual cues that can be used for making short-term predictions about a mobile agent's location without taking into account its goals or intentions. While knowledge of an agent's intentions may also be useful in making such predictions, that approach is not the focus of this study. Instead, we have implemented a neural network that takes as input a set of terrain features in the vicinity of a mobile agent, and with this information it generates a probability vector that predicts the likelihood that the agent will travel in each of fifteen different directions. This is transformed into a prediction about the agent's future location, along with a time period that it is valid. We integrated the prediction capability with the perceptual system of a virtual human pilot, which was implemented in Soar, an architecture commonly used for building knowledge-based agents.

2. VISUAL PERCEPTION IN A SYNTHETIC WORLD

This method was developed for virtual humans that are used in military simulations where combat situations are modeled at the level of individual entities [4]. An individual entity can be a truck, tank, aircraft, individual combatant, or any other distinct, active object. These entities are called *synthetic forces* and they are typically semi-autonomous, meaning that they require human intervention when they get stuck in situations where they do not know what to do next. Our efforts are focused on the creation of *intelligent forces,* which both achieve a significantly higher degree of autonomy and simulate human behavior [8,9,18].

Visual perception in our synthetic world involves three distinct problems: perception of terrain, perception of vehicle instruments, and perception of other entities. The first problem, the perception of terrain, is critical for tasks such as flying, especially at low levels where the possibility of colliding with the ground requires the pilot to continually monitor for the presence of obstacles in the flight path. The terrain is available on demand in the form of a polygonal surface. Major features such as rivers and roads are annotated in the terrain database and are available by querying by location. The virtual pilot perceives the terrain via a look-ahead sensor that samples points along the flight path for the altitude and the pilot adjusts the flight parameters accordingly [18]. This approach to terrain perception provides sufficient functionality for the pilot to fly safely, but it does not provide the level of realism needed for tasks such as flying along a contour line around a hill instead of simply flying over it. Such behaviors require a more general visual capability than we have yet addressed.

The second problem, the perception of cockpit instruments, is what enables the virtual pilot to keep track of the helicopter's current state (e.g.., air speed, altitude above ground level, heading, and so on). We do not model the eye gazing at the cockpit instruments, rather, the vehicle's state is continually perceived. Based on these parameters, the virtual pilot modifies the parameters of the helicopter for flying.

Finally, the perception of entities is necessary for tasks such as tracking, formation flying, and targeting. The virtual pilot perceives other entities using a simulated visual sensor, which is designed to model human visual perception of entities. Entity perception is driven by the arrival of a stream of entity-state updates. Each update characterizes the momentary state of an entity: it provides information such as its identity, location, and velocity. These updates are filtered through models of the pilot's visual sensors to determine what information is potentially perceptible. Entities that are too far away will be imperceptible. Entities within the perceptible range of the model may still be rendered imperceptible if they are occluded by a terrain feature or by an environmental factor such as smoke or dust. The sensor model also determines the resolution of the percept based on factors like distance, dwell time, and visibility. Hence, an entity may initially be recognized only as a vehicle when perceived at a great distance, but it may be identifiable as a specific tank model at a closer range.

3. VISUAL ATTENTION

Our ultimate goal is to create virtual humans with believable behaviors. One of the characteristics of humans is the ability to direct their visual attention to objects in the environment. This capability not only has an observable behavioral manifestation, but it also has implications for what an agent knows or does not know, based on what it has perceived. So for the purpose of achieving believable behavior, it is important to model visual attention. As it turns out, there is a functional need for visual attention in the synthetic world that mirrors the real world. Early models of our virtual pilot would crash their helicopters when there were a lot of entities in their visual field. Since they could see everything within a 360-degree arc and a radius of 7 kilometers, it should not have been too surprising that the perceptual-cognitive system would get bogged down since equal attention was at times being given to hundreds of objects in the environment simultaneously. What was lacking was a way for the pilot to focus its visual attention [8]. Attention filters out excess information and enables the perceiver to focus on a limited set of objects or region of space.

The motivation for *anticipating where to look* stems directly from the need to model visual attention in virtual humans, particularly when visual perception is reduced from a 360-degree field of view (FOV) to just 30 degrees. While it is well known that humans can typically perceive objects within a 210-degree horizontal field of view, the details are perceived in a much smaller area of the fovea, which measures 1-2 degrees across. Rather than represent the different levels of acuity in the visual field and all that this would entail, we chose instead to model a functional field of view of 30 degrees. The rationale behind this decision is straightforward. For this study we wished to develop a method for shifting attention from one moving object to another, where the reacquisition of an object is aided by a short-term prediction of its location in the context of complex terrain. Since objects are automatically recognized in the simulation, there is no need to model the saccadic eye movements that rapidly move the high-acuity fovea around an object during the recognition process. The functional field of view represents an area where human attention could reasonably be applied on a time scale of approximately one second or more for tasks such as search and tracking. The size and use of a functional field of view is comparable to what others have used. For example, Aasman [1] modeled automobile drivers with a functional field of view of 20 degrees and Reece [15] uses a 60 degree FOV for simulated soldiers.

Reducing the size of the field of view had the effect of filtering out a lot of stimuli, but it also forced the issue of how to control the pilot's focus of attention and gaze. For example, when a pilot is tracking two objects, one of which is not currently in view, it has to shift its visual attention between the objects, and it has to do it frequently enough to remain sufficiently aware of the situation to avoid disasters such as collisions. But enabling the pilot to shift its visual attention raises the question that is the focus of this paper: how long can it look away from a moving target without losing track of it? When it is time to shift visual attention back to a target, there has to be a reasonable prediction of where the object will be located to ease the reacquisition.

Now that we have presented the context for modeling visual perception, we will describe a method for predicting a target object's future position, and the amount of time the prediction is valid, given the current environmental state (e.g., terrain features) and the observed motion of the object.

4. APPROACH

The basic assumption behind this approach is that terrain influences the path taken by a vehicle, regardless of its destination. The goal is to model this influence as a predictive function that can be used to generate expectations about where an observed vehicle will move next. A simple, commonly used approach is the least squares method, which uses the history of motion to predict future behavior by extrapolating a path to fit the already established linear pattern. This approach can work satisfactorily in situations where the path is a trajectory of some sort, but when there are obstacles in the way of the vehicle, a recent history of motion may not produce valid predictions. Typical real-world terrain contains sharp turns in the road, major bodies of water, and mountain ranges that will influence the path taken by the vehicle. Figure 2 shows an example of a tank following a curved road. The tank's path is tightly constrained by the terrain—even if it chose to leave the road the mountains and the lake would serve as obstacles to be avoided. We also considered using Kalman filters [19] to make the predictions but decided against it for the same reason. Recursive estimation procedures are applied to information from the past to make a prediction, which does not take into consideration what can be seen in the future. Finally, we also considered using a rule-based approach to making these predictions, but decided against it since it would be too hard initially to acquire the complex relationships between terrain and movement.

In the end, we chose to use an artificial neural network to predict the near-term direction and location of a vehicle. The obvious advantage of neural networks is that they can be trained to recognize this relationship. Once trained, the artificial neural network can be used to look at the terrain in front of a vehicle and make a prediction about its near-term direction and location. The remainder of this section describes our modeling decisions, how the neural network was trained, and how the output of the neural network is transformed into a prediction about location.

4.1 Modeling the perception of terrain

For this study we chose to model the influence of seven terrain features: mountains, hard roads, soft roads, passable water, impassable water, buildings and forests. All of these features were available in the terrain database—by querying a specific location in the terrain one can find out whether one or more of these features is present there. Because of this way of storing information, we had to take samples of the areas of interest rather than doing an exhaustive query of the space.

Our algorithm takes samples of the terrain features along radials within the other agent's 120-degree field of view, every 15 degrees, out to the 600-meter range limit. Each radial is divided into three segments: the first is 100 meters long, the second is 200 meters, and the third is 300 meters. 20 samples are taken within each segment of each radial, thus samples are taken every 5 meters for the first 100 meters, every 10 meters in the second segment, and every 15 meters for the third segment. In all, the field of view is divided into 27 subfields (9 radials X 3 segments per radial), with 20 samples per subfield, for a total of 540 samples. Each sample results in a 7-tuple of boolean values, one for each terrain feature. If a feature is detected in one of a subfield's samples it is then considered to be present in the subfield. In the end, each subfield is represented as a 7-tuple, in the same was as the sample. Figure 2 shows the sampling pattern as a set of radial lines extending out from the nose of the target agent, oriented on the agent's current heading.

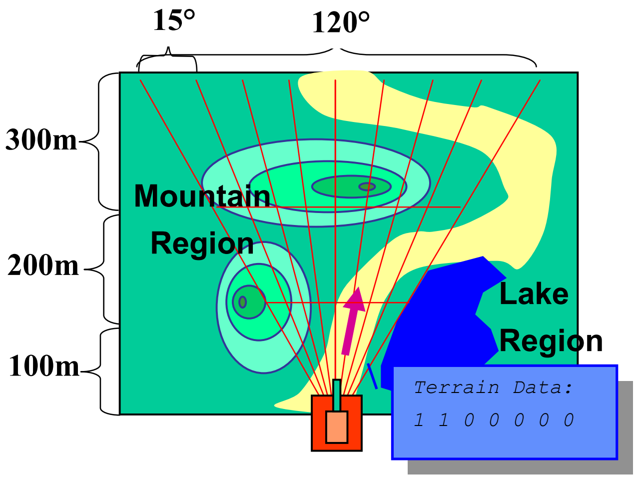

Figure 2: Modeling what another agent can "see"

4.2 Learning to predict the influence of terrain

We used the standard approach to training a neural network—we gave it sufficient training data to set the weights within the network and then applied the network to actual data to test its accuracy. To collect the training data, we ran a series of twenty different scenarios. In each scenario there was a single mobile agent, a tank, who either drove on a road or cross-country. The tank's highest speed was 48 km/hr (13.3 m/sec), and we assumed that the speed of the tank remained fairly constant on straight roads.

The neural network takes 190 inputs and produces 15 outputs. The inputs are delivered as a vector of boolean values; each of the 27 subfields contributes its own 7-tuple, which adds up to 189. In addition there is one bias input, making a total of 190 inputs to the network.

The output of the network is a vector of probabilities for each of 15 discrete headings that the agent may choose relative to its current direction, ranging from –35 degrees to +35 degrees, in five degree increments. If the output is 0 degrees it means that the agent will continue on its current heading. The value assigned to each angle indicates the probability that the agent will choose that particular direction.

4.3 Predicting where and when to look

It is not enough to predict the direction that a mobile agent will go in the next instant—we want to know where it will be located several seconds in the future, and we want to know how long the prediction will be valid. We set an upper bound of seven seconds on the forward projection since predicting further ahead does not appear to be cognitively plausible. Moray says that the limits of memory are a major factor in determining when visual attention should be next directed to an information source and seven seconds is a close approximation of the time between samples [11].

The algorithm shown below produces a vector of predicted locations for the observed mobile agent, along with the time period for which the prediction will hold. The predictions are in 1 second increments. The 1st tuple gives the predicted (x,y) location of the agent after 1 second, and the last tuple gives the predicted location after N seconds, where N will never exceed 7 seconds.

The steps for making the prediction are as follows:

1. N=1 (iteration #)
2. Observe the target object
    a. note the speed and location
    b. query the terrain database (take samples starting at agent's actual or predicted location)
3. Feed the terrain data to the neural network
4. Neural network outputs *probability-vector* with probability values for each of 15 discrete headings (-35, -30, -25, …, +35 degrees)
5. Calculate *decay-factor* (0.03 X N)
6. IF MAX(*probability-vector*) – *decay-factor* ≥ 0.80
    a. choose heading with MAX probability
    b. calculate *predicted-location* from speed, heading, and 1 second elapsed time
    c. N=N+1
    d. go to step 2b and use *predicted-location*,

    ELSE return {<*predicted-location*, 1>, …,
    <*predicted-location*, N>}

Figure 3: Prediction algorithm

The algorithm makes predictions by iteratively feeding samples of the terrain from in front of the mobile agent through the neural network, beginning with the mobile agent's current location, and then using the predicted locations to retrieve more terrain data for the network. Each time the neural network produces a probability vector for each of 15 discrete headings, and from this vector the heading with the highest probability is chosen, which will be used to calculate a another *predicted-location*. The iteration goes on until a confidence threshold has been reached. The confidence threshold is a measure of how much we trust the prediction—this measure is affected by the probabilities produced by the neural network and by a *decay-factor* that is used to guarantee that the algorithm terminates after 7 iterations (which covers 7 seconds).

5. INTEGRATING PREDICTION WITH PERCEPTION AND COGNITION

We developed the prediction method just described for a virtual pilot implemented in the Soar cognitive architecture [13]. Soar executes the decision cycle shown in Figure 4. In this framework, perception occurs during the input phase—this is when percepts are processed and the results are placed in the agent's working memory. During the elaboration phase, productions are matched with the contents of working memory and fire in parallel until quiescence is reached, meaning that no more rules fire. The rules that fire during the elaboration phase do not actually change the contents of working memory, rather, they create preferences for changes to working memory and they produce motor commands. These commands are issued to the motor system during the output phase. During the decision phase, a procedure evaluates the preferences that were generated during the elaboration phase; it decides what changes to make in working memory, and it chooses an operator to apply in the current context. Once the decision phase ends, the decision cycle begins again.

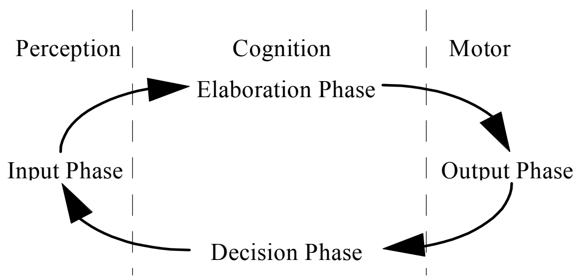

Figure 4: Soar decision cycle

The interface between cognition and perception in Soar occurs in working memory, where the percepts are placed during the input phase. Once in working memory, the percepts are matched and used by operators to perform tasks associated with the current goal hierarchy. The sensors are controlled by issuing commands to the motor system, completing the cycle between perception, cognition, and motor behavior.

Soar does not place constraints on the number of percepts that can be placed in working memory at one time. Nor does the theory behind Soar tell us how to control the level of detail produced by the perceptual system or how to control the amount of information processed during the input phase. As mentioned previously, the lack of a method for focusing attention would sometimes result in too much information being processed per decision cycle, seriously impairing the ability of the agent to cope with real-time tasks. By limiting the field of view, the information overload was reduced, but this introduced the problem of needing to handle divided attention, particularly when tracking mobile agents. It was in this context that we integrated the neural network and prediction algorithm. Each decision cycle during the input phase, the algorithm supplies Soar's cognitive system with predictions about the future location of selected individual mobile agents.

Cognition chooses which objects the perception system should attend to according to the goals and tasks that are being performed at any given time. When attention is shifted away from a target, an operator initiates the prediction algorithm for that vehicle if it is anticipated that attention will be needed there again. The predicted locations of the object being mentally tracked are placed into working memory like other percepts during the input phase. The predictions do not extend beyond seven seconds, so there is a natural limit to how long a pilot should comfortably look away.

If a task involves tracking multiple moving vehicles, the cognitive system receives input on the vehicle(s) that are in the field of view, and receives predictions about the vehicles it chooses to mentally track while they are not in view. After a set amount of time, an operator in the cognitive system can choose to shift the pilot's attention away from the current focus and back to the vehicle that it is mentally tracking. The pilot aims its visual sensor toward the predicted location of the target vehicle and if it successfully reacquires the target, then a visual tracking operator will keep it focused. If a target is not reacquired, then it will become necessary to perform a visual search, beginning in the region where it expected to find the object. We have not yet implemented this capability.

With the prediction data, the virtual pilot can look away from a mobile agent for up to 7 seconds, or whatever value N has in the prediction vector, and then look back to the predicted location from its own location, which may have also changed in the intervening period. If the pilot decides to look back sooner than the maximum time, it can use the prediction corresponding to the amount of time that has passed since last tracking the object.

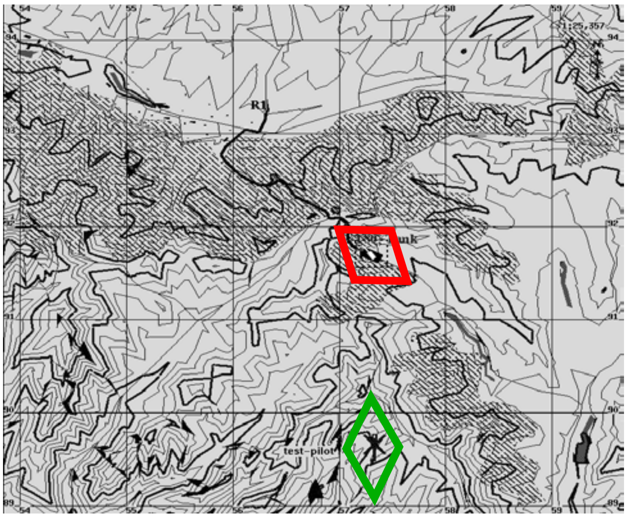

Figure 5: Road scenario

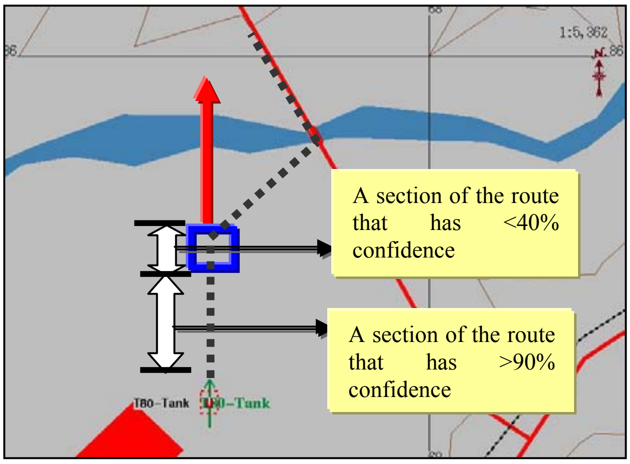

Figure 6: Cross Country Scenario

6. RESULTS

Figure 5 shows a scenario where a tank is following a road through a mountain area with forests. The algorithm accurately predicts that the tank will follow the road. When we made a comparison of the actual and predicted locations over time, the points showed very little deviation and corresponded to the road shape. This means that the road terrain feature affected the tank's movement and the pilot was able to accurately predict this movement.

Another general class of scenarios involves situations where the tank is travelling cross-country (i.e., it is not following a road). Such a case is shown in Figure 6, where the tank has come close to a river and must now find a bridge in order to cross it. As the tank drives within 300 meters of the river (see box) the predictability goes way down, as indicated by the shorter white arrow. This is because the terrain database does not provide bridge information, so the algorithm recognizes that the river will moderate the tank's behavior, but cannot predict how. The dashed line in Figure 6 indicates the path the tank will eventually take. With a shorter prediction time it becomes necessary for the pilot to track the mobile agent more closely until it is in a place where predictions can be made more confidently.

We ran the algorithm on mobile agents moving at different speeds to test the accuracy of the predictions. The mobile agent's speed is assumed to be constant for any given prediction. Table 1 shows the maximum error between the predicted and actual positions.

Table 1. Target Speed versus Error

Speed of the target object	Error
10 km/h	± 4 meters
20 km/h	± 8.5 meters
30 km/h	± 10 meters
48 km/h	± 15 meters

One of the weaknesses of our approach is a direct result of the computational cost of "perceiving" the terrain features, which involves querying a terrain database. Due to the real-time nature of the task, we had to selectively sample the terrain for specific features, but this can lead to missing critical information, particularly when the features are small. Bridges, buildings, roads, and rivers could all potentially be missed in any given sample set, assuming that such features were available. On the other hand, many different samples are typically taken as the mobile agent moves, so there is a cumulative likelihood that they will be captured at some point. But this approach still lacks the kind of coverage that the human visual system appears to have.

7. RELATED WORK

Many researchers in robotics and computer vision have worked on active vision or variants of this concept [3,14] but very few have addressed the issue of mental tracking as it relates to visual attention. From this body of research we considered two other approaches to making predictions about the location of mobile agents: (1) the least squares method, and (2) Kalman filtering [19]. Both of these techniques have been used extensively in computer vision, but the problem with them is that they do not take into consideration terrain features and their effect on the movement of vehicles. For example, a least squares prediction is shown in Figure 6 as the arrow showing a straight-line projection—its predictions are based entirely on past history, not current context.

Researchers concerned with creating virtual humans have addressed the issue of visual attention in a variety of ways. Rickel and Johnson created STEVE, an intelligent tutor and coach for an immersive environment [16]. STEVE acts like a human—he can direct his gaze to the student, to task-related objects, or to objects the student is manipulating. STEVE, however, perceives everything regardless of where the gaze is pointed, so prediction is not needed, just gaze control for the sake of believability. Chopra-Khullar's [5] Automated Visual Attending (AVA) system is one of the best existing overall models for generating believable gaze behaviors in an animated human character. With respect to the work described in this paper, AVA also makes predictions about where objects will appear when they are occluded from view, but it appears to use a linear model. Thus it can predict trajectories but not the effects of complex terrain.

Henninger et al. [7] did some similar work to ours on predicting vehicular movements using neural networks. They modeled road marches by giving the neural network the speed of the vehicle and waypoints in the road, but they did not model other aspects of the terrain.

Baluja and Pomerlau [2] used artificial neural networks to visually monitor and control a robot vehicle that follows a lane on a highway.

The psychological, cognitive science and computational neuroscience literature on visual attention and visual tracking is extensive. Findlay and Walker [6] developed a detailed model of saccadic eye control. Their model suggests that gaze is controlled by a complex interaction of spatial and temporal pathways in the brain, but it does not suggest how predictions are made, particularly on this time scale. Itti and Koch [10] have developed a saliency map that predicts what parts of a scene will capture bottom-up attention based on features such as color, orientation, contrast, and motion. Combined with some form of top-down control this approach shows promise as a method for controlling attention, although it does not account for projected movement through complex terrain yet.

8. CONCLUSIONS

This paper describes an approach to making short-term predictions about the movements of mobile entities in complex terrain. Terrain influences movement—cars generally follow roads rather than driving cross-country, and tanks do not drive directly over steep mountains or ford rivers even though that might be the shortest route to a destination. We trained an artificial neural network to make predictions about location up to seven seconds in advance and successfully integrated this information with the perceptual system of a virtual helicopter pilot in a military simulation. The underlying motivation for making such predictions is to enable virtual humans to more easily reacquire a moving target after shifting perceptual attention away from it in situations where there are multiple targets and divided attention.

There are still a number of issues that need to be addressed in future work. The research described here provides the capability of predicting where a target will be in cases where attention is divided or where visual contact is temporarily lost. We described how Soar operators use this information to switch between targets, but what is lacking is a general method for dividing attention. We have demonstrated that it is possible, but how should it be handled on a routine basis? We plan to start addressing this question by studying the goal hierarchies and tasks that our virtual humans perform and looking for patterns associated with balancing concurrent tasks with the need for perceptual monitoring. Chopra-Khullar's [5] dissertation describes an approach to scheduling perceptual tasks that we will take into account when designing the next phase of our system.

Another issue that needs to be addressed is how to integrate a routine method for handling divided attention with other mechanisms associated with visual attention, such as the integration of top-down and bottom-up control. We are looking closely at the work on saliency maps by Itti and Koch [10]. While their initial work focused on the bottom-up salience effects on attention, we are beginning to work with Itti on developing a task map to represent the effects of tasks on top-down attention. The task map will take inputs from the current task operators, the dialogue, aural events, as well as from the emotional state of the virtual human. The basic idea is that task map will make some perceptual objects more salient than others, which would influence the way that percepts are filtered and where the eye sensor is focused. In the end, our goal is to build a general framework for perceptual attention for virtual humans (see [17] for an example of this framework) that are capable of performing a wide range of tasks as well as holding conversations and other social interactions.

9. ACKNOWLEDGMENTS

This paper was developed with funds of the Office of Naval Research under grant N00014-98-1-0132 and the Department of the Army under contract number DAAD 19-99-D-0046. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the Office of Naval Research or the Department of the Army.

10. REFERENCES

[1] Aasman, J. (1995). Modelling driver behavior in Soar. Ph.D. Dissertation, Rijksuniveriteit Gronigen, Leiden, Netherlands.

[2] Baluja, S. and Pomerlau, D. (1997). Expectation-based selective attention for visual monitoring and control of a robot vehicle. *Robotics and Autonomous Systems*, Vol. 22, No. 3-4, December, 1997, pp. 329-344.

[3] Ballard, D. (1991). Animate vision. *Artificial Intelligence*, 48, 57-86.

[4] Calder, R.B., Smith, J.E., Courtemanche, A.J. Mar, J.M.F., Ceranowicz, A.Z. (1993). ModSAF behavior simulation and control. *Proceedings of the Second Conference on Computer Generated Forces and Behavioral Representation* (pp. 347-356). STRICOM-DMSO.

[5] Chopra-Khullar, S. (1999). Where to look? Automating certain visual attending behaviors of human characters. Ph.D. Dissertation, Computer and Information Science, University of Pennsylvania.

[6] Findlay, J. and Walker, R. (1999). A model of saccade generation based on parallel processing and competitive inhibition. *Behavioral and Brain Sciences,* 22, 661-721.

[7] Henninger, A., Gonzalez, A., and Georgiopoulos, M. (1999). Neural network based semi-automated forces: experimental results. *Proceedings of the I/ITSEC,* Orlando, Florida.

[8] Hill, R., (1999). Modeling Perceptual Attention in Virtual Humans. *Proceedings of the 8th Conference on Computer Generated Forces and Behavioral Representation*, Orlando, FL.

[9] Hill, R., Gratch, J., Rosenbloom, P., (2000). Flexible group behavior: Virtual commanders for synthetic battlespaces. *Proceedings of the Fourth International Conference on Autonomous Agents*, Barcelona, Spain, 2000.

[10] Itti, L and Koch, C. (2000) A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, Vol. 40, No. 10-12, pp. 1489-1506, May 2000.

[11] Moray, N. (1995). Designing for attention. In A. Baddeley and L. Weiskrantz, editors, *Attention: Selection, Awareness & Control*. Clarendon Press: Oxford.

[12] Neisser, U. (1967). *Cognitive Psychology*. NY: Appleton-Century-Crofts.

[13] Newell, A., (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

[14] Reece, D. and Shafer, S., (1995). Control of perceptual attention in robot driving. *Artificial Intelligence*, 78, 461-505

[15] Reece, D. and Wirthlin R. (1996). Detection models for computer generated individual combatants. *Proceedings of the Sixth Conference on Computer Generated Forces and Behavioral Representation,* Orlando, Florida.

[16] Rickel, J. and Johnson, W.L., (1999). Animated agents for procedural training in virtual reality: perception, cognition, and motor control. *Applied Artificial Intelligence,* 13:343-382.

[17] Swartout, W., Hill, R.., Gratch, J., Johnson, W.L., Kyriakakis, C., Labore, K., Lindheim, R., Marsella, S., Miraglia, D., Moore, B., Morie, J., Rickel, J.,Thiebaux, M., Tuch, L., Whitney, R.. Toward the holodeck: Integrating graphics, sound, character and story. *Proceedings of 5th International Conference on Autonomous Agents*, Montreal, Canada, June 2001

[18] Tambe, M., Schwamb, K., Rosenbloom, P. (1995). Building intelligent pilots for simulated rotary wing aircraft. *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation,* Orlando, Florida.

[19] Welch, G. and Bishop, G. (1995). An introduction to the Kalman filter. University of North Carolina, Department of Computer Science, TR 95-041