Object Persistence for Synthetic Creatures

Damian A. Isla   Bruce M. Blumberg

MIT Media Lab
77 Massachusetts Ave. NE18-5th fl.
Cambridge, MA 02139-4307
1-617-452-5603

{naimad, bruce}@media.mit.edu

ABSTRACT

We present methods for anticipatory behavior in simulated graphical creatures. We discuss in general terms the importance of anticipatory behavior through explicit expectation formation. We present an in-depth description of a specific type of expectation-formation, namely location-expectation, or object persistence. A new representation – the Probabilistic Occupancy Map (POM) – is presented, and it is shown how this representation can be used to maintain estimations of the positions of mobile objects in the world based on both positive and negative knowledge provided by the creature's perceptual system. Finally a number of illustrative results are presented that show Duncan, our simulated dog, successfully performing a number of tasks that require a high degree of spatial common sense.

Categories and Subject Descriptors

I.2.0 [**Artificial Intelligence**]: General – *cognitive simulation*

General Terms

Algorithms, Design, Theory.

Keywords

Autonomous agents, graphical agents, object persistence, spatial common sense.

1. INTRODUCTION

As we attempt to build increasingly sophisticated autonomous interactive agents (or "synthetic creatures" as we will call them in this paper) an important contributor to a life-like appearance is the ability to *anticipate*. Anticipation – or expectation formation – can be thought of as the ability to make decisions and react to aspects of the world state that, for one reason or another, cannot be directly perceived. This might include events that occur outside the field of view, are occluded, or that have not yet happened (but are *expected* to happen). There are many potential sources for these expectations, such as reliably observed event correlations (when the button is pushed, the elevator doors are expected to open), theory of mind (given an assumed state of mind, another creature might be expected to perform action A), physical intuition (when a ball is released, it is expected to fall), or, as will be discussed at length here, spatial structure (if a ball rolls behind a wall, it is expected to come out again after a certain delay). Not only do these types of expectations make a creature seem more intelligent, but their absence significantly *impairs* any pretensions it might have to common sense. If on two recent occasions a character got hit in the head with brick upon opening a window,

Figure 1: Duncan the Highland Terrier

then it would look either broken or colossally stupid for the character to open the window a third time without showing any kind of anticipation of being hit again.

Our group has, over the past few years, experimented with various aspects of expectation in graphically embodied creatures such as Duncan the Virtual Terrier (Figure 1). Duncan lives in a graphical environment which he perceives through a synthetic perception system which includes simulated audition and point-of-view rendering (synthetic vision). Duncan is a platform for various aspects of our research, including models of operant and classical conditioning, motor learning. In the work described in this paper, Duncan was given a sense of object persistence which allows him to maintain reasonable models of the locations of mobile objects in the world.

Section 2 is a brief discussion of expectation theory in general. Section 3 discusses in depth the problem of location-expectation. Section 4 introduces Probabilistic Occupancy Maps (POMs) as a means of maintaining location-expectation distributions. Section 5 presents some example results and concluding remarks are given in Section 6.

2. EXPECTATION THEORY

[8] provides an excellent overview of expectation theory. This discussion will largely parallel its conclusions.

In general, it is an observation of some sort – perhaps of an unusually salient stimulus – that leads to an expectation formation by a specialized *predictor* unit. These expectations are accompanied by a degree of *confidence*. The validity of an expectation is ascertained through *verification*. The process of verification can have three possible outcomes:

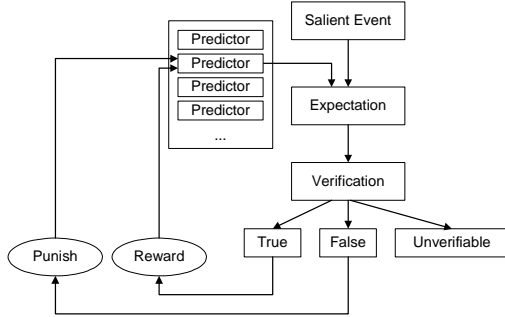- Verifiably true: The expectation turned out to be correct.

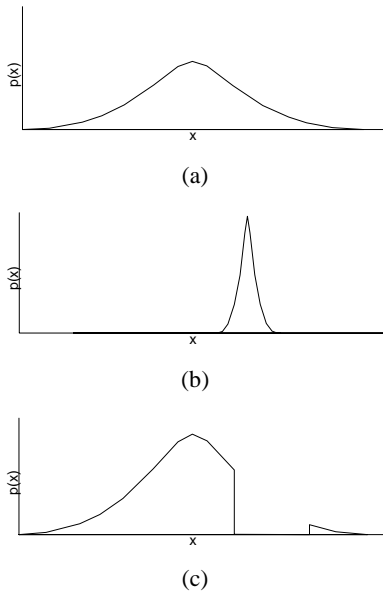**Figure 2: The expectation/verification loop**

(a)

(b)

(c)

Figure 3: Incorporating positive and negative observations.

A distribution (a) represents a space of predictions with the indicated confidence. An observation of the true state (b) results in a tightening of the distribution while the negation of some region of the state space (c) zeroes the distribution in that region, and scales the rest of the distribution up as appropriate.

- Verifiably false: The expectation turned out to be incorrect. This is an expectation violation.

- Unverifiable: The accuracy of the prediction could not be determined.

Typically, a verifiable prediction-outcome is followed by some form of belief revision. The predictor that generated the expectation might also be revised, for example, its reliability rating might be modified according to whether its outcome proved accurate or not. If the prediction accuracy was unverifiable then neither the expectation or the predictor itself is impacted. This entire process is summarized in Figure 2.

The above formulation applies to discrete true/false predictions. Some expectations, however, are better formulated as a *space* of predictions that might naturally be expressed as a probability distribution over the possible world states, as in Figure 3. Within this space, each infinitesimal element can be considered an individual prediction that can be verified as true or false. This space of predictions is coupled in two ways: first, in that an observation and confirmation of one prediction is equivalent to a negation of all the other predictions, resulting in a "tightening" of the distribution around the observed value (Figure 3b). Second, a verified negation of a subregion of the state-space (effectively a culling of probability from the distribution) results in a renormalization of the rest of the distribution (Figure 3c).

2.1 Expectation Violations and Salience

We define salience as the degree to which an observation violates expectation. As [8] points out there are two types of expectation violation: unexpected observations and negated expectations (verified false predictions). For the first kind, a straightforward inverse relationship exists between salience and confidence at the observed value (thus unexpected observations are more salient):

$$s(x) = \frac{1 - c(x)}{c(x)} \tag{1.1}$$

This maps salience into the range [0,Inf] (assuming that *c(x)* has range [0,1]). We call this form of salience *surprise*.

Expectation violations of the second kind can be considered proportional to the amount of confidence culled by the observation. This might be expressed as

$$s(x) = \int_R c(x) \tag{1.2}$$

where *R* is the region of the state-space of *x* that has been verified as false (see Figure 3c). We call this form of salience *confusion*.

It is notable that many secondary emotions imply some form of expectation: fear, delight, disappointment, anticipation, confusion, dread, worry, etc. They also imply an extended range of possible interactions. Creatures that can form expectations can be tricked, teased, deceived and more.

3. OBJECT PERSISTENCE

In this section we introduce an extended example of the utility of expectation-formation: that of location-expectation. We argue that the ability to maintain reasonable location-expectations is tantamount to a sense of object persistence. Object persistence, as discussed by the psychologist Piaget [Piaget, 1954 #15], refers to the persistence of mental images of objects after they have stopped being perceived. It also implies the ability to make basic deductions about where objects could be, and to act on those deductions. If a child turns toward a toy that has been hidden from it, it doesn't forget about the toy, or remain staring at the toy's last known position. Instead it searches for the toy systematically, based on its last observed position and based on the physical structure of the environment – could it have been hidden behind the box, or under the table? The basic problem of object persistence, which we would like our synthetic creatures to be able to solve, is: given that an object is not currently visible, where does the creature *presume* that object to be?

3.1 Distribution Representation

In forming probability distributions over space, we must come up with some way of representing those distributions. One natural
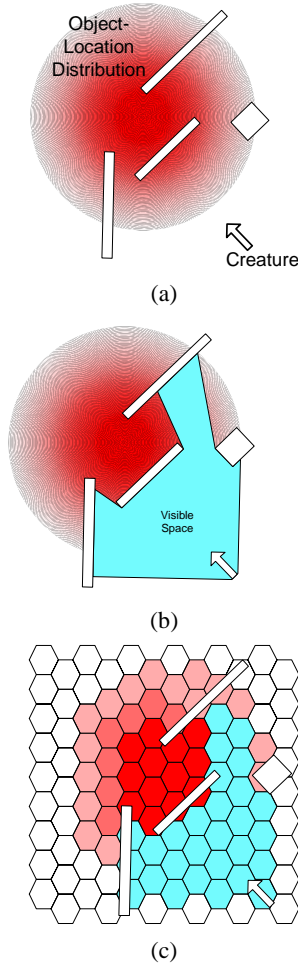
Figure 4: Distribution Representation

A gaussian distribution (a) cannot incorporate negative information about regions of space observed to be empty (b). In this work, we discretize the space itself, and allow each element to become a "bucket" of probability.

representation might be the gaussian distribution – after all, it is compact, and mathematically very convenient, with notions of confidence and "best guess" built in.

The difficulty with the straight gaussian representation is that it does not lend itself easily to the incorporation of negative knowledge. Figure 4 shows an example of this problem. Figure 4a, shows the distribution for an occluded object. A great deal of negative information – namely, those locations observed to be empty – is being thrown out. The correct distribution, the one in which all probability has been zeroed in all visibly empty space, is shown in Figure 4b. This distribution is clearly not gaussian – it is even disjoint. Since the shape of the distribution relies ultimately on the physical structure of the environment, it can be arbitrarily complex – and arbitrarily messy.

3.2 Probabilistic Occupancy Maps

The general strategy we use to overcome the difficulties of the gaussian representation is to discretize the location-distribution, through the use of a Probabilistic Occupancy Map (POM). In this

formulation the environment is partitioned into discrete locations which can be used as "buckets" of probability. Thus the compact gaussian representation is replaced with a vector of discrete activations, in which each activation corresponds to the probability that the target object is contained in that location. As Figure 4c shows, this scheme has no problem representing oddly-shaped distributions, though its accuracy depends on the resolution of the map itself.

We use a hexagonal grid overlaid on the environment as the general form of the map. Each node is connected to six neighbors, and is denoted by a pair of *(i,j)* indices (to maintain correct topology, the connectivity of these nodes alternates on even and odd rows). Each node has a world-space position, and represents all space for which it is the nearest node. Each node is also annotated with a probability vector, each element indicating the probability that the node contains a particular object. For simplicity, in the following discussion we will assume that there is only one target object to be tracked, and that each node is annotated with a single probability, $p(n_{i,j})$, of containing that object.

In some experiments the maps themselves were learned, such that higher resolution was used in more "interesting" parts of the environment, such as around prominent landmarks. This strategy is very useful in terms of representational economy. However, it is not essential to the basic implementation, and so will not be discussed here. See [6] for details.

3.2.1 Probability Diffusion

On timesteps in which the target object is not observed, a discrete diffusion step is carried out to reflect the decreased confidence in the target's location.

Simple isotropic diffusion works well. The update expression for a single map element, or node, can be given as

$$p^{t+1}(n) = (1-\lambda)\,p^t(n) + \frac{\lambda}{6} \sum_{n' \in neighbors(n)} p^t(n') \qquad (1.3)$$

where λ is a diffusion constant in the range [0,1] and $p^t(n)$ is the probability of the node n at time t. Thus at each timestep, each node passes some fraction of its own activation to its neighbors. The diffusion constant should be roughly proportional to the mobility of the object (such that an object that has been observed to be stationary has a low diffusion constant, and so on).

When the target was last observed in motion, the probability should diffuse preferentially in the direction of the target's last observed velocity. This can be used in the example in which the target object disappears behind an occluder – the creature can form an expectation about when it should reappear. The most obvious way to achieve this is to modify the diffusion rates so as to favor diffusion in the correct direction.

Assuming that a target object's velocity can be estimated through simple averaging of position changes over recent timesteps, we might set the diffusion rate at a specific node as

$$\lambda_i = \lambda_c + \max\left(0, \frac{v \cdot l_i}{|l_i|^2}\right) \qquad (1.4)$$

where v is the velocity vector, l_i is the position offset between the current node and the node's ith neighbor, and λ_i is the diffusion rate along the ith connection. λ_c is a constant diffusion rate, ensuring that some probability is diffused to every neighbor, even if that neighbor does not lie in the direction of the velocity vector.

At each timestep, therefore, each node gives λ_i of its total activation to each of its i neighbors, and keeps the remaining activation for itself. If the sum of these rates is more than 1, then the rates are normalized to add to 1 (since a node cannot give away more probability than it has). Note that if the velocity v is zero, then the entire expression is equivalent to equation (1.3).

To keep distributions from shooting off into infinity, we also slowly decay the velocity vector over all timesteps in which the object is not observed.

3.2.2 Observation
In timesteps in which the target object was observed, we pick the node $n*$ that is closest to the target's position. We tighten the distribution around this node by setting its probability to 1, and zeroing the probabilities of all other nodes:

$$\forall n_{i,j} \neq n^*, p(n_{i,j}) \leftarrow 0 \qquad (1.5)$$

$$p(n^*) \leftarrow 1 \qquad (1.6)$$

As in section 2.1, we determine a salience for the observation with the expression

$$s = \frac{1 - p(n^*)}{p(n^*)} \qquad (1.7)$$

While this gives a valid mapping for a salience in the range [0,Inf], it has one undesirable property: that when the distribution is very diffuse, even the most likely location (the location with the highest p^*) returns a high salience if confirmed. Since the desired behavior should be for the most expected prediction not to be considered salient when confirmed, we normalize the previous expression by $p_{highest}$, the probability of the most likely location. Thus,

$$s = \frac{p_{highest} - p(n^*)}{p(n^*)} \qquad (1.8)$$

3.2.3 Verification
Verification in this domain requires us to use information about locations observed to be empty. If a map node with a certain probability is considered a prediction about an object's location, then the observation of that location without the corresponding observation of the object can be considered a negation of that prediction. In such a case, the probability content of the node is zeroed. Once all visible locations are zeroed, the remaining distribution in non-visible locations is renormalized.

On timesteps in which the target object was not observed, the map nodes are divided into a visible set, V, and a non-visible set, N. First a total amount of culled probability p_{culled} is calculated as

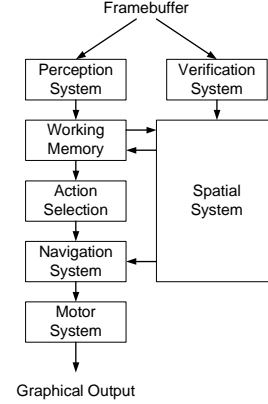$$p_{culled} = \sum_{n \in V} p(n) \qquad (1.9)$$
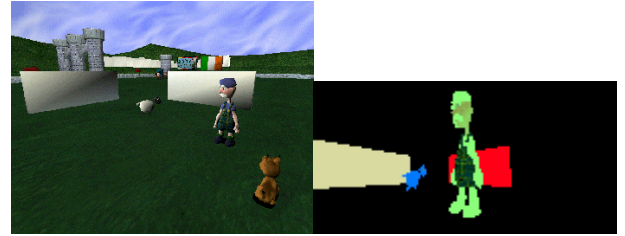

Figure 5: C4 System Overview

Figure 6: Duncan's Synthetic Vision

Then the probabilities of all visible nodes are set to zero:

$$\forall n \in V, p(n) \leftarrow 0 \qquad (1.10)$$

Finally the remaining probabilities are renormalized

$$\forall n \in N, p(n) \leftarrow \frac{1}{1 - p_{culled}} p(n) \qquad (1.11)$$

Note that by the definition given by equation (1.2), the value p_{culled} is also the measure of the surprisingness of the failure to observe the target object.

4. IMPLEMENTATION
The POM scheme was implemented on top of C4, a character simulation platform described in [7] and [2]. Figure 5 shows a basic overview of the system architecture, with the POM integrated as a *Spatial System*. Currently, Duncan (Figure 1) is our most sophisticated character, and it was in his brain that most of the following was implemented and tested.

4.1 Synthetic Vision
A simple model of synthetic vision was used as the primary source of perceptual information for Duncan. Similar to schemes used in [1] and [18]. The vision model consisted of a rendering of the world from the point of view of Duncan's left eye. This rendering was color-coded (as shown in Figure 6) such that individual objects were recognized by color (i.e. no shape-analysis was performed). From these renderings, screen-space object-centroids were extracted, and combined with the contents of the depth-buffer to produce eye- and then world-space coordinates. Two implications of this strategy are that an object is

considered visible if *any* portion of that object is visible, and that the location is taken as the centroid of the visible portion.

4.1.1 The Test-point Method

Another crucial function of the vision system is to determine visibility of map-locations for the purposes of negative verification (i.e. in separating the *N* and *V* sets, from section 0). A number of schemes are possible here, the "truest" one probably involving pixel-sampling over the expected screen-space extent of the node. This scheme is not used due to run-time considerations.

Instead, we use a vastly simplified *Test-Point* approximation, in which a single point is imagined floating half a body-height above the world location of the map-node to be tested. The world-coordinates of this point are transformed back into NDC coordinates. The depth-buffer is then queried to see whether an object at that point would be visible. If the depth of the pixel to which the test-point transformed is less then the NDC-depth of the test-point, then the test-point is hidden, and the entire location is considered non-visible (and placed in the *N* set). If the depth is greater, then the location is considered visible (and placed in the *V* set).

Unlike object-observation, in which the view of any part of the object counts as a full observation, it is generally advantageous in location-visibility to err on the side of non-visibility. In other words, it is advantageous to demand that the *entire* location be visible before the corresponding node is considered visible. The danger is that if the target object is in a location that is considered visible without being observed, then its true location will be discounted as a possible location. On the other hand, the risk of mistaking a visible location for a non-visible location is only that a location the creature is not at might be considered for a longer period than otherwise, especially since the likely reaction will be to approach that location and examine it further.

4.2 Working Memory and Object Matching

Duncan maintains a Working Memory model that consists of a recent perceptual history of all objects in the world. On each timestep, the perceptual input from each object – color, shape and location – from the vision system is bundled up into a *Belief* object. These beliefs are then compared to a list of persistent beliefs stored in the creature's Working Memory. Since many of the new beliefs are probably a new observation of an old belief (if it represents the newest data from an object that has been tracked over multiple timesteps, for example), all incoming beliefs undergo a process of *belief matching*, in which the "distances" between the new belief and each of the old ones is found. If the lowest of these "distances" is below a threshold then, the two are considered a match, and the data contained in the new belief is added onto the perceptual history contained in the old one. If the lowest distance is above the threshold, then the new belief is considered to represent a new object, and is itself added to Working Memory.

In finding the distance between two perception tuples, *<color, shape, location>*, each component in one tuple is compared with the corresponding component of the other. The total distance is then considered the sum of the component-distances. For the purpose of this system, both shape and color are symbolic binary matches returning distance infinity or zero depending on whether there is an exact symbolic match or not (e.g. "red" = "red").

In comparing location however, we can again make use of the location distributions of the POM. The incoming location data can be ascribed a nearest node, n^*. For each of the persistent objects, the POM maintains a separate location-distribution, in which n^* holds some probability. To find the distance between a location-distribution, and a new observation at n^*, we use

$$d = \frac{p_{highest} - p(n^*)}{p(n^*)} \qquad (1.12)$$

where $p_{highest}$ is again the highest probability value of any node in the map. This is, of course, simply a re-casting of the salience function of equation (1.8). This follows the intuition that if an observation is too unexpected, we might reject it altogether. In this case, if we were to observe an object in one location and then turn around and see what seems to be the same object in a radically different location, we might be more prone to think that it is a different object which looks exactly the same, rather than that the original object teleported from one place to the other.

When a new belief is merged into the old one, the appropriate distribution tightening, as per equations (1.5) and (1.6) occurs. For all non-visible objects, the appropriate distribution culling is carried out, as in equation (1.10), followed by the diffusion of all location-distributions, as in equation (1.3).

4.3 Action Selection and Motor Control

Having updated his perceptual image of the world, Duncan must decide what to do. Though the problem of Action Selection is a major focus of our research (particularly with regards to learning), it is largely irrelevant to this work. In many of the experiments conducted on this system, Duncan was given just three behaviors: "look at target", "approach target" and "approach shepherd" (the shepherd being a user-controlled character inhabiting the same space). Duncan decided between these behaviors deterministically at the instruction of the shepherd. These high-level behaviors are translated by the action selection system into low-level navigation and motor commands. A Navigation System was use to control locomotion toward a goal and obstacle avoidance, and a Motor System was used as the lowest-level interface with the graphics system to control Duncan's geometric joint-angles over time.

5. RESULTS

This section presents a few scenarios as examples of Duncan's new spatial capabilities with the POM system in place.

5.1 Salient Moving Objects

One result of the POM system is that moving objects –especially irregularly moving objects – automatically become salient. If an object remains still, its distribution looks like a roughly gaussian cluster of probability centered on the observed location of the object. When the object starts moving, the object is matched into a relatively unlikely part of this distribution – thereby making that observation salient. When velocity-based (anisotropic) diffusion is used, an object that zigzags is more salient than an object that moves at a constant speed in a constant direction.

5.2 Search Behaviors

There are a number of examples of "emergent search" that fall out of the POM system almost for free.

5.2.1 Emergent Look-Around

Intelligently moving the eyes over a visual scene, from an animation point of view, can be important in maintaining the illusion of life [16]. However, there is no practical point to this if the negative information about empty space is not used. Typically, the "look-around" behavior is a canned animation. However, a convincing "look-around" happens naturally with the POM system. Consider a situation in which a target (in this case, a sheep) has not been seen in some time (and has been moved since it was last observed), and is represented by a wide, diffuse distribution. When instructed to look at the sheep, Duncan turns towards the most likely position for the sheep, namely the center of the distribution. Not finding it there, he looks first one way, then the other, always looking at the current most likely position. The result is an alternating sweep-left, sweep-right behavior, until the sheep is found at its new location.

5.2.2 Emergent Search

If Duncan is instructed to approach the sheep, rather than just look at it, then the emergent look-around becomes an emergent search. As in the emergent look-around, Duncan approaches at every point the most likely location for the target object. He is preceded by his gaze however, which rules out open areas of the environment. This leaves only areas occluded by barriers, and the resulting behavior is what looks like a systematic search of these occluded areas.

In one example, Duncan is introduced to the environment shown in Figure 7. After a brief view of the environment and the objects in it, Duncan is called to the shepherd. Behind his back, the sheep is moved behind the nearest wall. When told to approach the sheep, Duncan immediately goes to search behind the correct wall. Duncan is called away again, and this time the sheep is moved from its original hiding place to a new one behind the other wall. When again instructed to approach the sheep, Duncan first looks in the original hiding place. Not finding it there, he comes out and again scans the scene, this time concluding that the only place the sheep can now is behind the second wall. Again, he finds the sheep in its hiding place.

What is perhaps most interesting about this example, is that at a behavioral level – at the level of the Action Selection mechanism – Duncan has only two behaviors to chose from: "approach the sheep" and "approach the shepherd". However, the result is much more sophisticated, and has the *appearance* of deliberative reasoning.

This "emergent search" behavior is extremely important for keeping the behavioral level clean. If "search" needed to be included at the behavioral level, it would need to be listed explicitly as a strategy for "approaching the sheep," "approaching the shepherd," "biting the sheep", "eating the sheep" etc. Matters would be even worse if we expected learning to occur at this level of control – thus pushing onto the learning mechanism the burden of deciding when "biting the sheep" necessitates a "search." With the architecture described here – in which some of the decision-making has been pushed up into the perceptual/memory system and down into the navigation/motor system – some degree of search capability is implied in *all* targeted behaviors.

5.3 Distribution-Based Object-Matching

To show the advantage of distribution-based object-matching, a scenario was set up as shown in Figure 8. Like in a previous

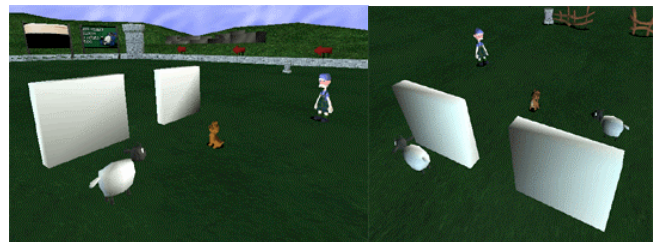**Figure 7: The search task environment**

Figure 8: The object-matching task environment

example, Duncan tracks a moving sheep as it passes behind a long wall. This time, however, the wall has a large gap in it. The scene is staged, because at a certain time where the sheep, unimpeded, might have expected to come out from behind the right wall, a second sheep, initially hidden, comes out from behind the right wall. A single sheep would have been observed to pass through the gap. Since it did not, the intelligent observer is forced to conclude that there are, in fact, two separate sheep. Happily, Duncan makes the same determination, since his constant viewing of the space between the two walls keeps any of the probability from diffusing from behind one wall to behind the other. Thus when the second sheep appears, it appears in a location for which the first sheep's location-distribution is zero. Since it is impossible for the two observations to represent the same object, the belief-matching step fails and a new persistent belief is added to Working Memory. This is apparent, because when Duncan is subsequently asked to approach the nearest sheep, he looks behind the left wall first.

This is an exciting result for two reasons. First, it shows a very different kind of "reasoning" from the previous examples – a kind of negative reasoning, reflecting the ultimate determination that the two objects are not the same. Second, it shows a kind of reasoning that is highly distributed: the vision system provides observation data, the Verification System extracts visible locations, the Spatial System maintains the POM itself, and the Working Memory system makes the ultimate decision to allocate a new persistent belief. In the best sprit of the Society of Mind [9], it is the confluent effect of all these systems that results in the "intelligent" behavior.

5.4 Emotional Behavior

Duncan maintains a list of autonomic variables to represent various aspects of internal state. Some of these variables are

explicit "emotional" variables. In one series of experiments, Duncan was given the variables of "surprise", "confusion" and "frustration". The first two of these were variables that autonomically decay over time. The values of equations (1.8) and (1.9) then provided offsets to these values. The "frustration" variable grew by some factor on every timestep that Duncan's object of attention was not directly observed. Figure 9 shows a trace of these variable values over time in a typical "search for the sheep" task.

Currently, the only effect that these variables have is to change Duncan's facial expression (as shown in Figure 10). However, the ultimate hope would be to feed these emotional variables back into Duncan's basic decision-making processes, such that, for example, frustration could act as a signal to "stop whatever you're doing and try something else", and surprise could indicate "examine more carefully your object of attention" etc.

6. CONCLUSIONS

In this paper, we have presented a model of object persistence and a representation – the Probabilistic Occupancy Map – that implements it. We draw a number of major conclusions.

Architecturally, the POM is an important lesson in multi-layer decision-making. The section on emergent search shows that even with a simple explicit behavior-selection layer, many complicated behaviors can result with the right perception/space-modeling systems. The behavioral level could be completely user-directed, or scripted, or could be run by a complex automatic behavior simulation system. Whatever the case, there is clear benefit to separating out this series of spatial competencies into separate layers.

Another important point is that a good representation is worth a lot. In this case, the POM structure was tuneable enough to be useful, but also general enough to account for many interesting effects. For example, it was fairly simple to incorporate a sense of momentum for velocity-based probability diffusion. It was also general enough that unforeseen behaviors emerged without being designed for.

Most importantly, this work is an illustration of the expectation theory that is presented in section 2. It should show that (a) representations based on this theory are conceptually simple and easy to implement, (b) the representations are powerful, and lead to interesting behaviors and abilities and (c) these new abilities contribute substantially to a creature's believability and apparent common sense. It is also hoped that this work can serve as a model for how other kinds of expectation formation abilities might be incorporated into a synthetic creature framework.

7. RELATED WORKS

The field of artificial life is relatively new, but already has such seminal works as [15] and [1]. More recently, work by Terzopoulos and his colleagues (e.g. [18] and [5]) has proved very impressive, and relevant to the current work for their concern with higher-level cognitive processes and cognitive modeling.

There has been work on expectation theory. [12] points out another source of surprise: events for which there were no expectations but that are inherently unusual. The example given by the author is that of a brick flying through a window: no prior expectation existed over where or when such an event would
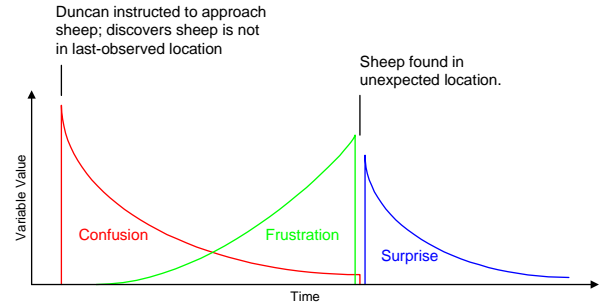

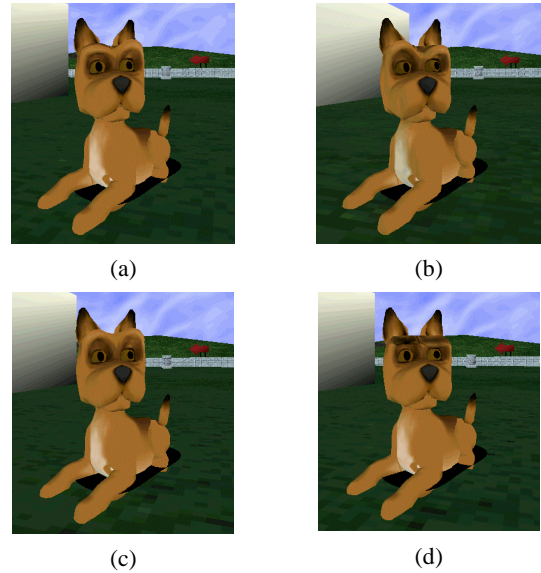Figure 9: Trace of emotional variables in a search task.

Figure 10: Duncan (a) neutral (b) confused (c) surprised (d) frustrated

occur, but it should nonetheless register a surprise. We do not treat this form of surprise in this work, though we believe that such models of inherent salience could easily be incorporated into this framework. As already mentioned before, [8] was very influential to this work.

Drescher did work specifically on cognitive processes leading to object persistence [3] though only in a toy world. His work was itself an exploration of the computational side of Piaget's theories (as in [13]).

Probabilistic Occupancy Maps are clear derivatives of the Occupancy Grids, or Evidence Grids used in the mobile robotics literature (e.g. [10]). The major difference is that the "probability" held by each spatial element in Occupancy Grids refers to the probability of that element being occupied, rather than the probability that it should contain any particular object. Occupancy Grids are not position-distributions over space, rather they are a method for sensor-fusion over space.

More generally, there has been a tremendous amount of work done on the modeling of emotions, such as [14], [19] and [11] to name but a few. Recent work on modeling emotional social agents has come out of the Synthetic Characters group itself, with [17]. In terms of emotion modeling, the current work seeks to go beyond the simple happiness/sadness models of emotion (e.g. [4])

to secondary emotions such as confusion or surprise that express slightly more subtle aspects of a character's internal state.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] B. M. Blumberg, "Old Tricks, New Dogs: Ethology and Interactive Characters,".: PhD. Thesis, MIT Media Lab, 1996.

[2] R. Burke, D. Isla, M. Downie, Y. Ivanov, and B. Blumberg, "CreatureSmarts: The Art and Architecture of a Virtual Brain," presented at The Game Developers Conference, San Jose, CA, 2001.

[3] G. Drescher, *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*: MIT Press, 1991.

[4] P. Ekman, "An Argument for Basic Emotions," in *Cognition and Emotion*, 1992.

[5] J. Funge, X. Tu, and D. Terzopoulos, "Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters," presented at Siggraph 99, Los Angeles, 1999.

[6] D. Isla, "The Virtual Hippocampus: Spatial Common Sense for Synthetic Creatures,".: Masters Thesis, MIT, 2001.

[7] D. Isla, R. Burke, M. Downie, and B. Blumberg, "A Layered Brain Architecture for Synthetic Creatures," presented at The International Joint Conference on Artificial Intelligence, Seattle, WA, 2001.

[8] C. Kline, "Observation-based Expectation Generation and Response for Behavior-based Artificial Creatures,".: Masters Thesis, Massachusetts Institute of Technology, 1999.

[9] M. Minsky, *The Society of Mind*. New York, NY: Simon and Schuster, 1985.

[10] H. Moravec, "Sensor Fusion in certainty grids for mobile robots," presented at Sensor Devices and Systems for Robotics, Catalonia, Spain, 1987.

[11] A. Ortony, G. Clore, and A. Collins, *The Cognitive Structure of Emotions*: Cambridge University Press, 1988.

[12] A. Ortony and D. Partridge, "Surprisingness and Expectation Failures: What's the Difference?," presented at Internation Joint Conference on Artificial Intelligence, 1987.

[13] J. Piaget, *The Construction of Reality in the Child*. New York: Norton, 1954.

[14] R. Picard, *Affective Computing*: MIT Press, 1997.

[15] C. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Computer Graphics*, vol. 21, 1987.

[16] F. Thomas and O. Johnson, *The Illusion of Life*: Hyperion, 1995.

[17] B. Tomlinson and B. Blumberg, "Social Behavior, Emotion and Learning in a Pack of Virtual Wolves," presented at AAAI Fall Symposium, 2001.

[18] X. Tu and D. Terzopoulos, "Artificial Fishes: Physics, Locomotion, Perception, Behavior," presented at SIGGRAPH, Orlando, FL, 1994.

[19] J. D. Velasquez, "Modeling emotions and other motivations in synthetic agents," presented at AAAI 97, 1997.