Computational Studies of Exploration by Smell

Wee Kheng Leow
Department of Information Systems and Computer Science
National University of Singapore
Lower Kent Ridge Road, Singapore 119260, Singapore
leowwk@iscs.nus.edu.sg

Abstract

Research on exploratory and searching behavior of animals and robots has attracted an increasing amount of interest recently. Existing works have focused mostly on exploratory behavior guided by vision and audition. On the other hand, research on smell-guided exploration has been lacking, even though animals may use the sense of smell more widely than sight or hearing to search for food and to evade danger.

This paper contributes to the study of smell-guided exploration. It describes a series of increasingly complex neural networks that each allows a simulated creature to search for food and to evade danger by smell. Other behaviors such as obstacle negotiation and risk taking emerge naturally from the creature's interaction with the environment. Comparative studies of these networks show that there is no significant performance advantage for a creature to have more than two sensors. This result may help to explain why real animals have only one or two smell sensing organs.

1 Introduction

To many animals, being able to detect distant food by smell is vital for their survival. The sense of smell can be more important than sight and hearing, especially for nocturnal animals. An understanding of how these animals search for food and evade danger by smell can be useful for industrial applications such as tracking the source of air pollution or sniffing out illegal drugs by a mobile robot. Over the years, research on computational models of exploratory behavior has attracted an increasing amount of interest (e.g., Arbib and House, 1987; Arbib and Lee, 1993; Mura and Franceschini, 1994; Webb, 1994). However, these works have focused largely on exploratory behavior guided by vision and audition. Computational studies of smell-guided exploration, on the other hand, have been lacking. Existing works on olfactory-motor coordination such as Preiss and Kramer (1986), Belanger and Willis (1996), Kramer (1996), and Willis and Arbas (1996) focus on modeling insects' odor-guided flight paths rather than exploratory behavior.

This paper contributes to the study of smell-guided exploration. It describes neural networks that allow a simulated creature to search for food and to evade danger by smell. The research objective is not just to construct *any* network that can perform the tasks, but to study *minimal* networks and their performance in the limit. As pointed out by Meyer and Guillot (1991), in order that we can understand what network architecture is important for solving what kind of behavioral problem, it is necessary to establish the minimal solution of a problem and to explore the limits of the solution.

Four types of neural networks that implement increasingly complex search strategies will be described and their performance will be compared under various conditions. An interesting surprise discovered in the simulations is that there is no significant advantage for a creature to have more than two smell sensors. This result may help to explain, from the computational point of view, why real animals typically have only one or two smell sensing organs.

Exploration guided by smell turns out to be a non-trivial task. To help the readers to better understand the difficulty of the task, this paper will first discuss in Section 2 some background information

To appear in *Adaptive Behavior*, 6(3–4), 1998.

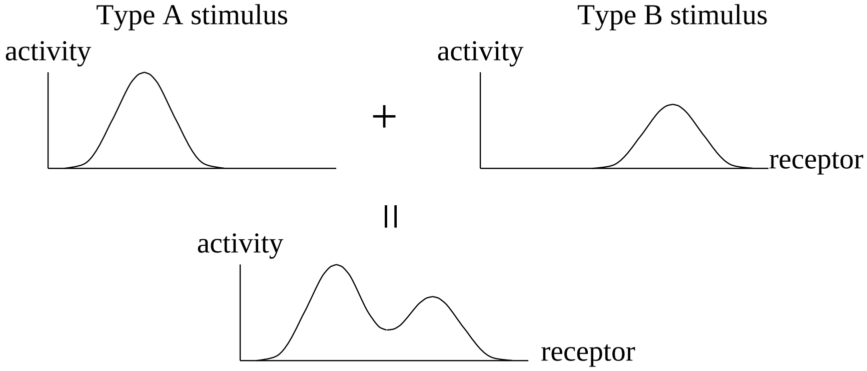


Figure 1: The receptors' activity pattern represents the odorant type and their activation levels encode the stimulus intensity. The receptors' response to a combination of odorants is assumed to be a linear combination of the responses to individual odorants.

regarding smell sensing. Next, in Section 3, the architectures and basic behaviors of the neural networks will be illustrated. Section 4 will compare the performance of the networks. In Section 5, complex behaviors that emerge from the creature's interaction with the environment will be illustrated. A summarizing discussion of the networks' behaviors and a comparison with related research works will be presented in Sectioin 6, followed by the conclusion.

2 Background

2.1 Sensing Odorants

The smell sensing organ, or smell sensor, of an animal consists of many olfactory receptors. Studies in neurophysiology (Kandel et al., 1991; Masson and Linster, 1995) show that the receptors' activities are determined by both the type and the strength of odorants. A receptor is broadly tuned to respond to a variety of odorants, but it responds most strongly to a preferred type of odorant. Taken as a whole, the receptors show different activity patterns in response to different odorant types, and the strength of pattern depends on the odorant intensity, i.e., the amount of air-borned chemicals present. The more the chemicals, the stronger are the receptors' activities.

In the presence of a variety of odorants, the smell sensor responds in a complex manner. Its response to a combination of odorants is not necessarily a linear combination of the responses to individual odorants (Masson and Linster, 1995). Nevertheless, it is assumed in this paper that such a linear combination is approximately correct (Fig. 1), so that essential network mechanisms required for food searching and danger avoidance can be easily identified. This assumption is justifiable since this work focuses on odor-guided exploration rather than odor discrimination.

This paper focuses only on exploration in a static environment in which food, danger sources, and obstacles are stationary. It is also assumed that there is no wind in the environment and that chemicals are diffused evenly in all directions. A simple model of such an environment is that the stimulus intensity at a particular location is inversely proportional to the squared-distance to the source.

2.2 Exploration by Smell

Finding target by smell is inherently more difficult than searching by sight. Visual information is directional because light travels in a straight path. A creature that detects the light emitted or reflected from a target can easily compute its direction to the target, and then approach it in that direction. On the other hand, olfactory information is local and non-directional. An odorant source disperses chemicals in all directions. The amount of chemicals at a particular location decreases with increasing distance from the source. A creature's smell sensors can only detect the type and the amount of chemicals at its present location, and this information is insufficient for it to locate the source.

There are many ways to obtain enough information to deduce where is the odorant source. The following is a list of possible methods in increasing amount of information and neural hardware requirement:

1. Temporal Difference. The creature can compare the odorant intensities sensed at the previous and current time steps. If the creature has moved a short distance since the previous time step, then it can decide its next move based on the temporal difference of stimulus intensity. For example, it can decide to continue moving forward if it senses an increase in odorant intensity. This method requires only one smell sensor and a memory of the stimulus intensity at a previous time step.

2. Spatial Difference. For a creature that has two smell sensors located at a significant distance apart, it can compare the odorant intensities detected by the two sensors and bias its movement towards the side with a stronger stimulus intensity.

3. Local Maximum. If a creature could have many sensors located around its body, then it could determine which sensor detects the largest amount of stimulus intensity, and then move along the direction represented by the sensor. This strategy is a generalization of the spatial difference method.

4. Gradient Ascent. With many sensors, a creature can also compute the local spatial differences of odorant intensity at a variety of directions and move along the direction of greatest increase in intensity. This method computes a second-order spatial difference of stimulus intensity.

Among these four search strategies, the temporal difference method requires the least amount of information and neural hardware. It, therefore, represents a minimal neural network capable of exploration by smell. The next section describes neural networks that implement these search strategies.

3 Network Architectures

The neural network that implements temporal difference method will be discussed first since it is the simplest network capable of exploration by smell. The networks for the other search strategies are extensions of the basic architecture.

3.1 Temporal Difference Network

The temporal difference network is divided into three main modules: the *Sensor Module*, the *Detection Module*, and the *Motor Control Module* (Fig. 2). The Sensor Module detects the presence of chemical stimuli in the environment. It consists of two sets of neural units denoted by a_j and $\bar{a}_j$. Each of the chemical receptors a_j is broadly tuned to a variety of odorants, and its activity is proportional to the amount of chemical present. The sensor's response to a mixture of stimuli is a linear combination of the responses to the individual stimuli (Fig. 1). The units $\bar{a}_j$ computes the normalized activity of a_j:

$$\bar{a}_j = \frac{a_j}{\sum\limits_k a_k} \tag{1}$$

which will be used by the Detection Module to recognize the odorant types.

The Detection Module consists of two identical sub-networks, one for detecting food and the other for detecting danger (Fig. 2). The only difference between them is that their connections from units $\bar{a}_j$ in the Sensor Module carry different weights such that they detect the presence of either food or danger but not both. The weights are assumed to be pre-coded or learned from past experience. The sub-network for detecting food will be described below; the one for detecting danger will be omitted.

The presence of more than one stimulus complicates the architecture of the Detection Module. A creature that navigates based on the raw sensor activity may be tracking the combined effect of the stimuli rather than a particular stimulus. To be able to navigate towards one of the stimuli, say the dominant stimulus, the creature has to extract the intensity of the dominant stimulus from the sensor

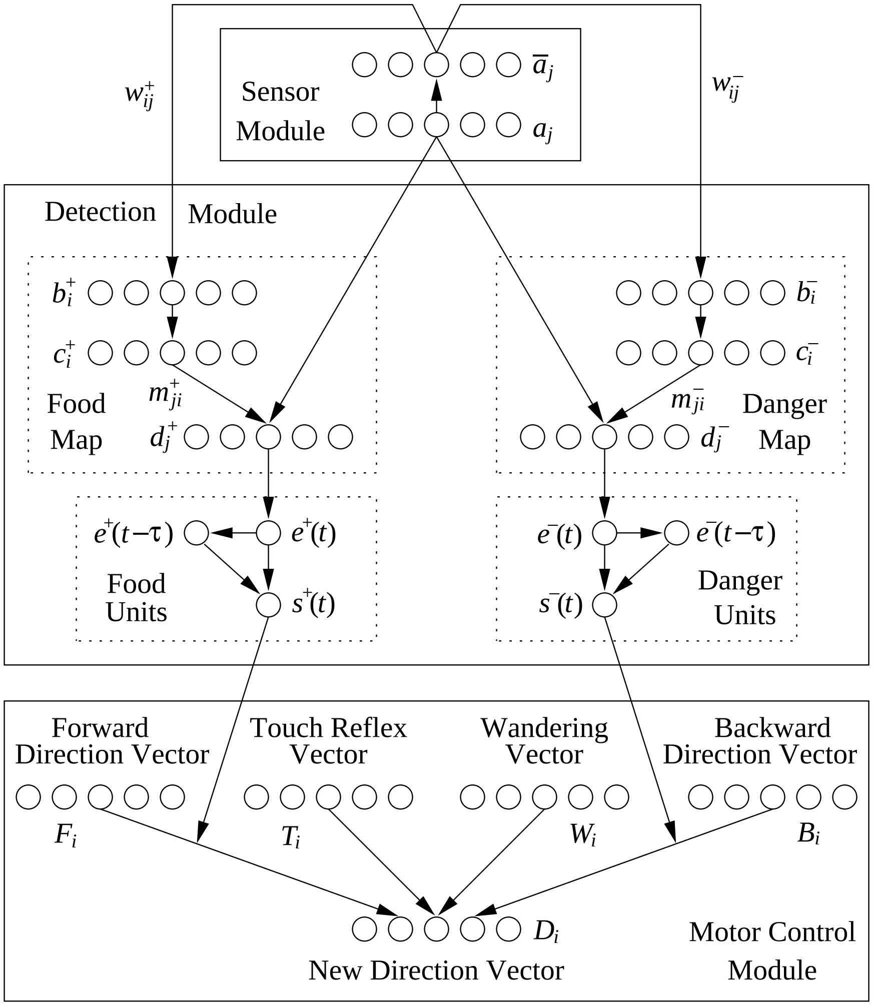


Figure 2: The sensorimotor coordination network is divided into three main modules: (a) the Sensor Module consists of olfactory receptors, (b) the Detection Module consists of Food Maps, Food units, Danger Maps, and Danger Units, and (c) the Motor Control Module consists of various Direction Vectors. See text for details.

activity that records the combined effects of the stimuli. This task is achieved by the *Food Maps* and the *Food Units* in the Detection Module.

Each Food Map unit b_i^+ detects the presence of one type of known stimulus by computing the weighted-sum of the inputs from units $\bar{a}_i$ (Fig. 3):

$$b_i^+ = \max\left(\sum_j w_{ij}^+ \, \bar{a}_j - \epsilon \ , 0\right) \tag{2}$$

where w_{ij}^+ is the weight of the connection from unit $\bar{a}_j$ to unit b_i^+ and ϵ is a positive constant threshold representing the creature's *sensitivity* to stimulus intensity. The threshold ϵ is currently set at 0.001 so that the creature can detect small amounts of stimulus intensity. Since the peak stimulus intensity at the source is taken as 1, the current value of ϵ means that the creature is sensitive to stimulus intensity as small as 0.1% of the peak intensity. The smaller the value of ϵ, the more sensitive is the creature's response. The weights w_{ij}^+ encode the sensor's activity pattern in response to stimulus i. In other words,

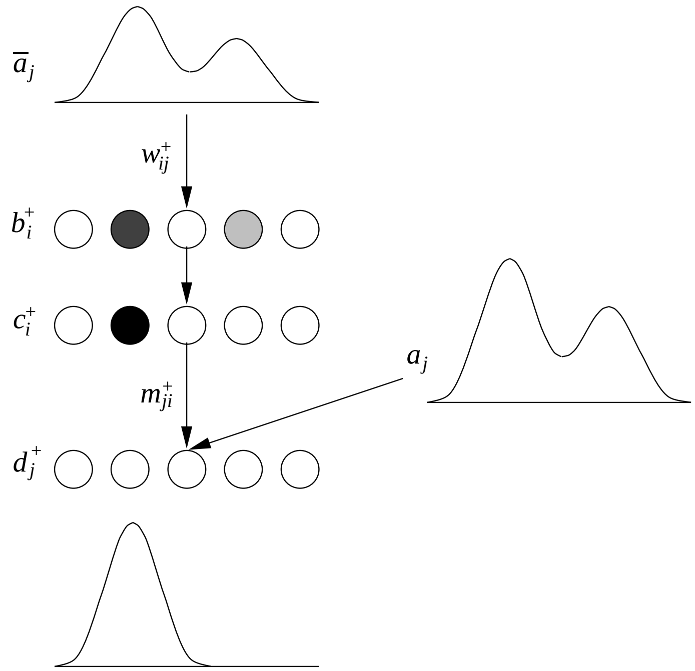


Figure 3: The Food Map extracts the dominant stimulus from the combined activity pattern in the smell sensor. It consists of 3 layers of units: b^+_i, c^+_i, and d^+_j. Units b^+_i are activated to different levels (denoted by different shades of grey) depending on how well their weights w^+_{ij} match the activity pattern in $\bar{a}_j$. Units c^+_i determine the winner (black unit) which corresponds to the dominant stimulus type. The winner projects the activity pattern of the dominant stimulus into units d^+_j, weighted by activities a_j. Units d^+_j now encode the type and intensity of the dominant stimulus.

the weights encode the stimulus that unit b^+_i recognizes. The weights are normalized such that

$$0 \leq w^+_{ij} \leq 1 \text{ for each } i, j \quad \text{ and } \sum_j w^+_{ij} = 1 \ . \tag{3}$$

Unit b^+_i will be strongly activated if the stimulus that it recognizes is present.

The Food Map units c^+_i form a winner-take-all network which determines which of the units b^+_i is most active:

$$c^+_i = \begin{cases} 1 & \text{if } b^+_i = \max\limits_j b^+_j \\ 0 & \text{otherwise} \end{cases} \ . \tag{4}$$

At most one of the units c^+_i is active at any one time and it identifies the dominant stimulus. The weights m^+_{ji} of the connections from unit c^+_i to unit d^+_j carry exactly the same values as w^+_{ij}. The single active unit c^+_i projects the activity pattern corresponding to the dominant stimulus into the units d^+_j, weighted by sensor activities a_j:

$$d^+_j = a_j \sum_i m^+_{ji} \, c^+_i \ . \tag{5}$$

Units d^+_j now encodes the type and the intensity of *only* the dominant stimulus (Fig. 3). In other words, information regarding the dominant stimulus has been extracted from the sensor inputs which may contain information about several stimuli.

The activities of units d^+_j are summed up by the Food Unit $e^+(t)$

$$e^+(t) = \sum_j d^+_j(t) \tag{6}$$

to extract the intensity of the dominant stimulus (see Appendix A for a detailed explanation of how this is accomplished). The Food Unit $e^+(t-\tau)$ receives the activity of unit $e^+(t)$ with a time delay of τ, thus keeping track of the intensity of the dominant stimulus detected at the previous time step. The Food Unit $s^+(t)$ measures the increase in stimulus intensity over one time step, i.e., the temporal difference of stimulus intensity:

$$s^+(t) = \max\left(e^+(t) - e^+(t-\tau), 0\right) . \tag{7}$$

The Motor Control Module integrates the information extracted in the Food and the Danger maps to determine the creature's new heading (Fig. 2). In this Module, the *New Direction Vector* D_i encodes the new egocentric direction of motion, i.e., the new direction relative to the creature's current heading. Each unit D_i is broadly tuned to encode a range of directions, but is most strongly activated for its preferred direction denoted by the unit vector $\mathbf{v}_i$. The resultant direction $\mathbf{D}$ encoded by the whole direction vector is the vector sum of each D_i's preferred direction:

$$\mathbf{D} = \sum_i D_i \mathbf{v}_i \ . \tag{8}$$

The advantage of this direction encoding method is that the vector sum of two directions can be easily computed by adding the activities of the corresponding direction representations. This encoding method has been observed in the Accessory Optic System of cats and primates (Grasse and Cynader, 1991).

In addition to the New Direction Vector, there are four other direction vectors that influence the creature's heading. The *Forward Direction Vector* F_i and *Backward Direction Vector* B_i are constant vectors that encode the forward and backward egocentric directions. The *Touch Reflex Vector* T_i generates a change of heading when the creature encounters obstacles. The creature has touch sensors (not drawn in Fig. 2) on its body to sense the presence of obstacles. When it detects an obstacle on the left side of its body, for instance, the Touch Reflex Vector T_i will generate a pattern encoding a direction towards the right. The exact direction encoded by T_i is random, but it has the largest probability at 45° towards the left or right of the current heading. If no obstacle is detected, T_i has zero magnitude. The random *Wandering Vector* W_i generates random bias towards either side of the creature's forward direction.

The summed effect of these direction vectors determine the creature's new heading:

$$D_i = \alpha^+ s^+ H F_i + \alpha^- s^- B_i + \beta\, T_i + \omega\, \epsilon\, W_i \tag{9}$$

where α^+, α^-, β, and ω are constant parameters, and ϵ is the sensitivity parameter discussed earlier (Eq. 2). The variable H represents the creature's internal state of *hunger level*, which is equal to 1 when the creature is not hungry and increases above 1 when the creature becomes more and more hungry.

The direction vectors F_i, B_i, T_i, and W_i have identical patterns (similar to the Gaussian function) except that they are centered at different directions. In particular, F_i is centered at the creature's forward direction, B_i at the backward direction, T_i at the direction away from the obstacle, and W_i at a random direction. The relative contributions of the vectors to the new heading D_i depend on their corresponding weighting parameters. In the current implementation, $\alpha^+ = \alpha^- = 1$. That is, given the same amount of appetitive (food) stimulus s^+ and aversive (danger) stimulus s^-, the Forward and Backward vectors will have equal influence on the new heading. The parameter β takes the value 10 so that when the creature touches an obtacle, the vector T_i can override other direction vectors. The Wandering Vector W_i is weighted by ϵ so that random wandering takes effect only when the creature detects little or no stimulus. Vector W_i is further weighted by the *randomness* parameter ω, which determines how strongly W_i affects the new heading D_i in comparison with other direction vectors. The value of ω ranges from 0 to 1, and can affect the network's performance quite drastically. This observation will be discussed in detail in Section 4.1.

In summary, Equation 9 embodies the creature's strategy in food searching, danger avoidance, and obstacle negotiation:

- When the creature senses an increase in appetitive stimuli (food) intensity, it is biased towards moving forward. The stronger the increase and the higher the hunger level, the stronger is the bias.

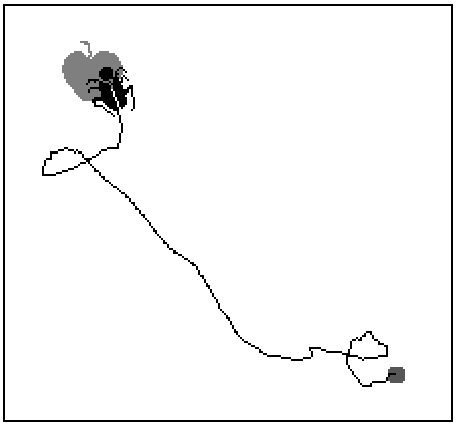


Figure 4: Searching for food using temporal difference strategy. The square dot denotes the starting position and the apple represents the food. The curve line indicates the path taken by the creature, which tends to be tortuous due to the effects of the random Wandering Vector.

- When the creature senses an increase in aversive stimuli (danger) intensity, it is biased towards turning around and moving in the reverse direction. The stronger the increase in stimulus intensity, the stronger is the bias.
- When the creature touches an obstacle, it is biased towards the direction away from the obstacle.
- At all time, the Wandering Vector biases the creature away from its current heading.

Although the above strategy looks simple, it is sufficient for the creature to locate food, evade danger, and overcome obstacles.

3.2 Basic Behaviors of Temporal Difference Network

The following simluation results illustrate the basic behaviors of the creature in food searching and danger evasion. Obstacle negotiation will be shown later in Section 5.1.

Figure 4 shows a typical path taken by the creature in search of a stationary food. Smooth sections of the path indicate places where the creature continuously detects increase in stimulus intensity. Sharp turns along the path indicate places where the creature detects no increase in stimulus intensity, and its heading is changed by the random Wandering Vector. Due the randomness of the Wandering Vector, the path taken by the creature tend to be tortuous.

Figure 5 shows that the temporal difference network can identify the dominant stimulus when the environment contains multiple stimuli. The food on the left produces a higher stimulus intensity than the one on the right. At the starting position, the difference in stimulus intensity is not significant enough and the creature moves along the direction of increase in *total* stimulus intensity. If the creature were unable to identify the dominant stimulus, it would have continued to move towards the middle of the two food sources. Instead, it moves directly towards the food on the left after identifying the dominant stimulus. Moreover, the path taken is more direct than that taken when there is only one food (Fig. 4).

Figure 6 shows a typical path taken by the creature in evading a stationary danger source. The fire symbol represents the danger source and the circle denotes the boundary of the danger zone within which the creature can detect aversive stimulus. Initially, the creature happens to fall inside the danger zone. It immediately turns around and moves out of the danger zone. Thereafter, when it again wanders across the boundary of the danger zone, it immediately turns around and moves away. As in food searching, the path taken in evading danger also tends to be tortuous due to the effect of the random Wandering Vector.

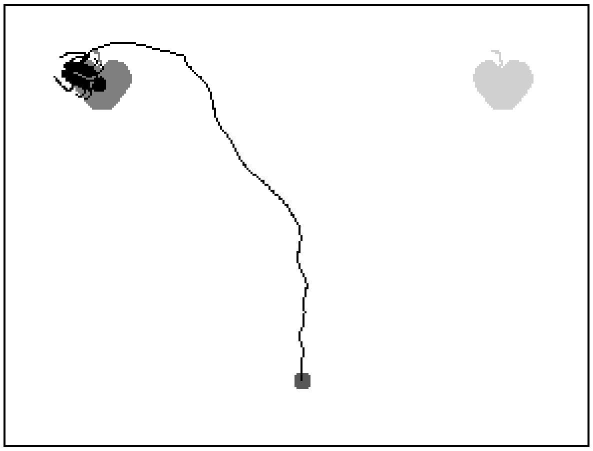


Figure 5: Searching for dominant food using temporal difference strategy. The food on the left produces higher stimulus intensity than the one on the right. The creature identifies the dominant stimulus and moves directly to it.

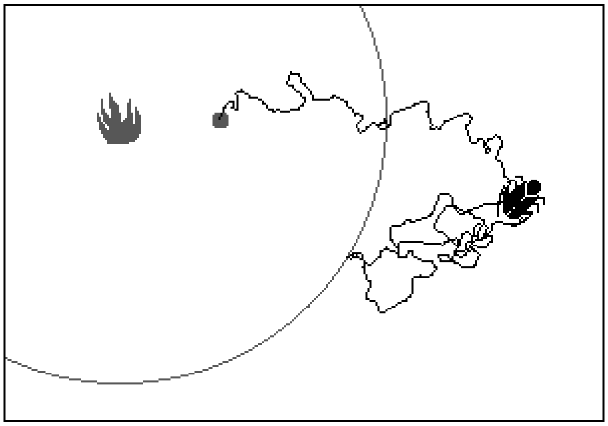


Figure 6: Evading danger using temporal difference strategy. The fire symbol represents danger source and the circle denotes the boundary within which the creature can detect aversive stimulus. The creature quickly moves out of the danger zone and avoids going in again.

3.3 Spatial Difference Network

The neural network that implements spatial difference strategy is similar to the temporal difference network, except that it consists of two smell sensors, one located on the left of the creature's head and the other on the right. In addition, it has two food maps and danger maps, connected to the corresponding smell sensors. The network equations of the sensors, food maps, and danger maps are identical to those of the temporal difference network. However, the equations for the food units and danger units are different—in this case, spatial difference strategy is used.

Let e_L^+ and e_R^+ denote the food units that sum the inputs respectively from the left and the right food map. Then, spatial difference is computed as:

$$s_L^+ = \max\left(e_L^+ - e_R^+ , 0\right) \tag{10}$$

$$s_R^+ = \max\left(e_R^+ - e_L^+ , 0\right) . \tag{11}$$

That is, unit s_L^+ is activated if the left sensor detects stronger stimulus intensity than the right sensor, and similarly for unit s_R^+.

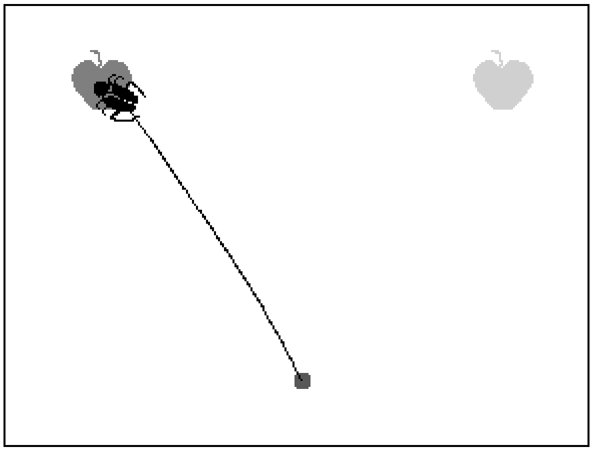


Figure 7: Searching for dominant food using spatial difference strategy. Compared to temporal difference strategy, the spatial difference method enables the creature to move more directly towards the dominant stimulus.

The Motor Control Module also differs from that of the temporal difference network. The spatial difference network does not require the Forward Direction Vector F_i and the Backward Direction Vector B_i. Instead, it has a constant *Left-Bias Vector* L_i and a constant *Right-Bias Vector* R_i. Except for vectors L_i and R_i, the spatial difference network has the same Touch Reflex Vector T_i and Wandering Vector W_i as the temporal difference network. The creature's new heading D_i is determined jointly by L_i, R_i, T_i, and W_i:

$$D_i = (\alpha^+ s_L^+ H + \alpha^- s_R^-) L_i + (\alpha^+ s_R^+ H + \alpha^- s_L^-) R_i + \beta\, T_i + \omega\, \epsilon\, W_i \ . \tag{12}$$

In essence, the creature is biased towards the side where more appetitive stimulus is detected, and away from the side where more aversive stimulus is detected. Most of the parameters are set to the same values as those of the temporal difference network: $\alpha^+ = \alpha^- = 1$, $\beta = 10$, and $\epsilon = 0.001$. With the presence of Left- and Right-Bias Vectors, the spatial difference network does not require the Wandering Vector to change the creature's heading. Therefore, the randomness parameter ω is set to a small value of 0.01 just for random wandering in the absence of stimulus.

Figure 7 illustrates the behavior of the spatial difference network in food searching. With spatial difference strategy, the creature takes a more direct route towards the food than the one that uses temporal difference method (Fig. 4). However, the path is not exactly a straight line—the creature actually makes small local turns along the path. In the same manner, the route the creature takes to evade danger is also more direct than that taken using temporal difference strategy.

3.4 Local Maximum Network

The network that adopts the local maximum strategy requires sensors all around the creature's body to detect stimuli intensities at different locations. In the current implementation, 8 sensors are used: 2 on the creature's head (as for the spatial difference network), 2 each on the left and the right side of the body, and 2 at the rear. All the sensors are equally spaced, and each is connected to its corresponding food maps, food units, danger maps, and danger units. Besides the 8 sensors and 8 sets of maps, the local maximum network has two additional units, S^+ and S^-, that sum up the activities of the food units s_i^+ and danger units s_i^- respectively.

In the Motor Control Module, a variable Forward Direction Vector F_i encodes the direction represented by the winning food unit s_i^+, and a Backward Direction Vector B_i encodes the direction *opposite* to that represented by the the winning danger unit s_i^-. The creature's new heading D_i is determined by the combined effects of the direction vectors:

$$D_i = \alpha^+ S^+ H F_i + \alpha^- S^- B_i + \beta\, T_i + \omega\, \epsilon\, W_i \ . \tag{13}$$

The creature is therefore biased towards the direction of maximum appetitive stimulus intensity, and away from the direction of maximum aversive stimulus intensity. The parameters are set to the same values as for the temporal and spatial difference networks. With more sensors, the creature has more information about the location of the food. As expected, it is able to take a direct straight path towards food and away from danger.

3.5 Gradient Ascent Network

Like the local maximum network, the network that implements the gradient ascent strategy also has eight smell sensors but it uses the stimulus information differently. Instead of computing the direction of local maximum stimulus intensity, it computes the direction of greatest increase in stimulus intensity. The network contains 8 additional units r_i^+, each computing the difference between the food unit e_i^+ on one side of the creature's body and the unit e_I^+ on the opposite side:

$$r_i^+ = \max\left(e_i^+ - e_I^+, 0\right) \tag{14}$$

That is, each unit r_i^+ computes the spatial difference of stimulus intensity along the I-to-i direction. Units r_i^+ are, in turn, connected to food units S_i^+ which form a winner-take-all network to determine which of the r_i^+ units win. Hence, the gradient ascent network computes a second-order spatial difference of stimulus intensity.

The Motor Control Module of the gradient ascent network is identical to that of the local maximum network. The creature is, therefore, biased towards the direction of greatest increase in appetitive stimulus intensity, and away from the direction of greatest increase in aversive stimulus intensity. The gradient ascent network behaves in a similar manner as the local maximum network in food searching and danger evasion.

4 Comparisons of Search Performance

The previous section illustrated four types of neural networks each implementing a different search strategy. Using spatial difference, local maximum, and gradient ascent methods, the creature takes direct routes towards food and away from danger. With temporal difference method, the creature's paths tend to be tortuous due to the strong influence of the random Wandering Vector.

This section compares the performance the search strategies in more details. Two experiments were carried out to study their performance:

1. Effect of randomness parameter ω on search performance.
2. Search performance in a noisy environment.

4.1 Effect of Randomness

The purpose of this experiment is to study how the randomness parameter ω affects the creature's performance in searching for food. The creature started off in the configuration shown in Fig. 4. Ten trials were run with different initial random seeds which altered the random turns that the creature made in each trial. The numbers of time steps taken to locate the food were recorded and averaged over the ten trials. These trials were repeated for each search strategy and for different values of randomness parameter ω. Experimental results are shown in Fig. 8. With local maximum and gradient ascent strategies, the creature located the food in the shortest amount of time. Using spatial difference method, the creature took a little more time to locate the food but its performance was comparable to those of the local maximum and gradient ascent methods. With temporal difference strategy, the creature took more than twice the amount of the time to locate the food compared to other search methods.

Figure 8 also shows that the randomness parameter ω has a marked effect on the performance of the temporal difference network and weaker effects on the other networks. In particular, the temporal difference network achieved its best performance with ω equal to 0.5. With smaller ω, the effect of

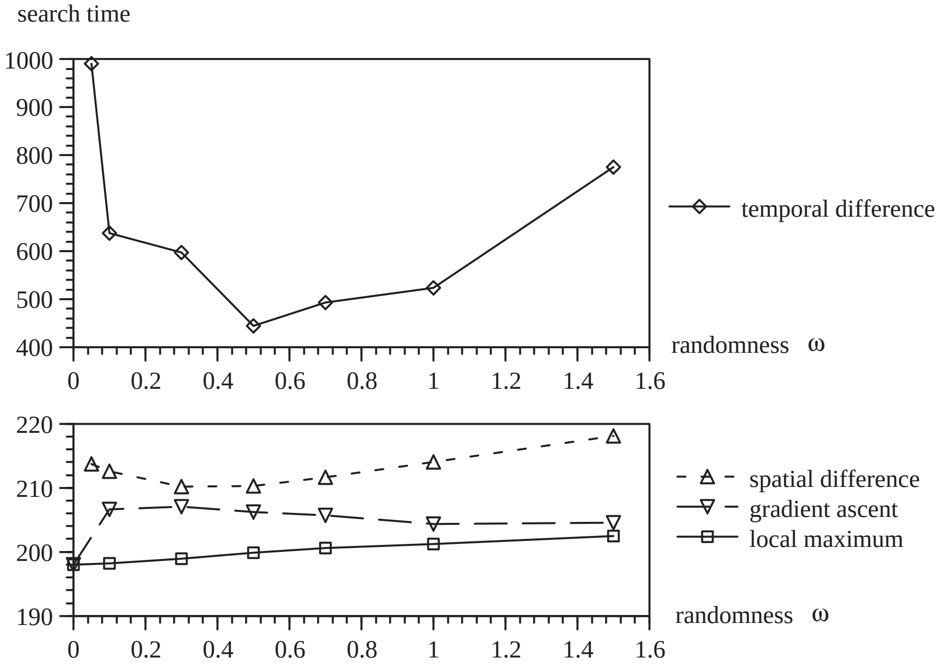


Figure 8: Effects of randomness on search performance. The randomness parameter ω has a marked effect on the performance of temporal difference network and weaker effects on the other networks. The temporal difference network achieves its best performance with $\omega = 0.5$.

the random Wandering Vector was too weak to turn the creature towards the food. As a result, the creature kept going around the food and could not close in onto the food quickly. With larger ω, the Wandering Vector's effect was too strong and it caused the creature's path to become erratic. Therefore, an appropriate amount of random wandering is very important for creatures that search for food using temporal difference method.

4.2 Noisy Environment

In the simulations performed above, the stimulus intensity at a particular location is inversely proportional to the squared-distance to the source. In the real world environment, however, such a law is never strictly obeyed due to air perturbation, dissipation of chemicals, etc. These factors that affect the stimulus intensity can be modeled as noise.

To study the networks' performance in an noisy environment, an experiment similar to the one described in the previous section was conducted. It differed from the previous experiment in that the randomness parameters were fixed while random noise was added to the stimulus intensity. The randomness parameters were set at 0.5 for temporal difference network, 0.01 for spatial difference network, and 0 for both local maximum and gradient ascent networks. As in the previous experiment, ten trials were run for each search strategy and for various noise levels.

Experimental results are shown in Fig. 9. The horizontal axis indicates the noise level measured as a percentage of the peak stimulus intensity at the source. For example, a noise level of 0.5 means that random noise with a magnitude of ±0.5% of the peak intensity was added to the stimulus intensity. Although 0.5% seems like a small percentage, it is actually very significant at large distances from the source. For instance, in this experiment, the stimulus intensity at the starting position is only about 10% of the peak intensity at the source. So, a noise level of ±0.5% at the source is equal to a noise level of ±5% at the starting position, which is a very significant amount.

Experimental results show, as expected, that the creature took longer to find the food as noise level increased. Temporal difference network's performance was most strongly affected by noise. The search

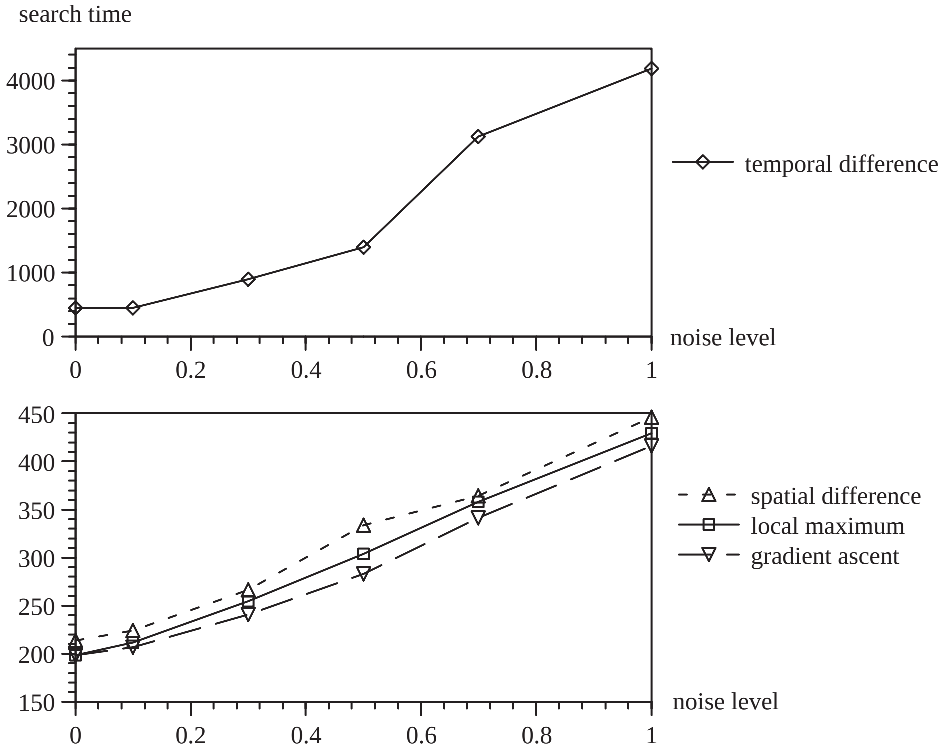


Figure 9: Effects of noise on search performance. Temporal difference network's performance is most strongly affected by increasing noise. Spatial difference network's performance is robust and comparable to those of the local maximum and gradient ascent networks at all the noise levels tested.

time increased from about 500 time steps at 0% noise to more than 4000 at 1% noise—an increase of about 8 times. On the other hand, the performance of the other networks was less strongly affected by noise. The search time increased from about 200 time steps at 0% noise to 450 at 1% noise—a little more than 2 times. This experiment shows that the performance of the spatial difference network is robust and comparable to those of the local maximum and gradient ascent networks.

5 Complex and Emergent Behavior

The network architectures described in Section 3 contain only simple components: smell and touch sensors, networks for computing temporal or spatial difference of stimulus intensity, and various direction vectors that bias the creature's heading one way or the other. There is no complex system for route finding or path planning. Despite the simplicity of the networks, complex behaviors emerge naturally from the creature's interaction with the environment. This section illustrates some of the complex behaviors including obstacle negotiation, danger avoidance, and risk taking. The four different networks behave in similar ways, and it suffices to illustrate just the spatial difference network's behavior.

5.1 Obstacle Negotiation

In the real world, obstacles can hinder the spread of airborne chemicals. To simulate such a situation, the stimulus intensity at a position behind the obstacle is set at a lower value than it would if there were no obstacle. Figure 10 shows how the creature looks for food while negotiating obstacle. When the creature encounters the obstacle, the touch sensors on the right side of its body fires causing it to make a left turn. As it moves on, it continues to make left turns whenever the right side of its body touches the obstacle. Eventually, it clears the obstacle and then takes a direct path towards the food. This

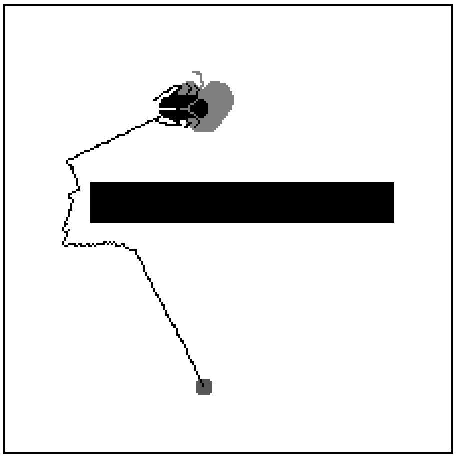


Figure 10: Negotiating obstacle while searching for food using spatial difference strategy. The creature skirts around the obstacle and then moves directly towards the food.

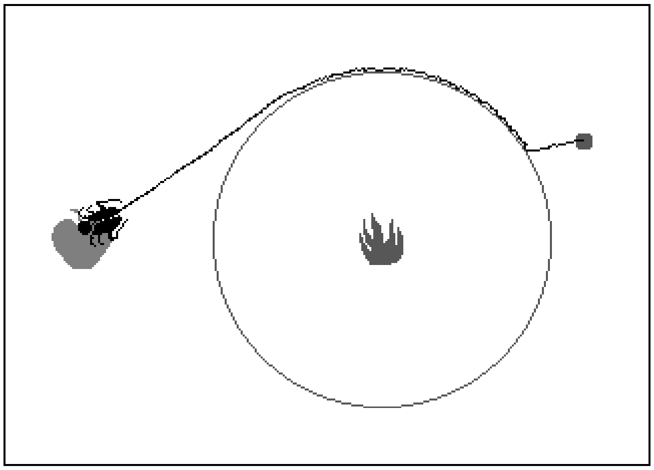
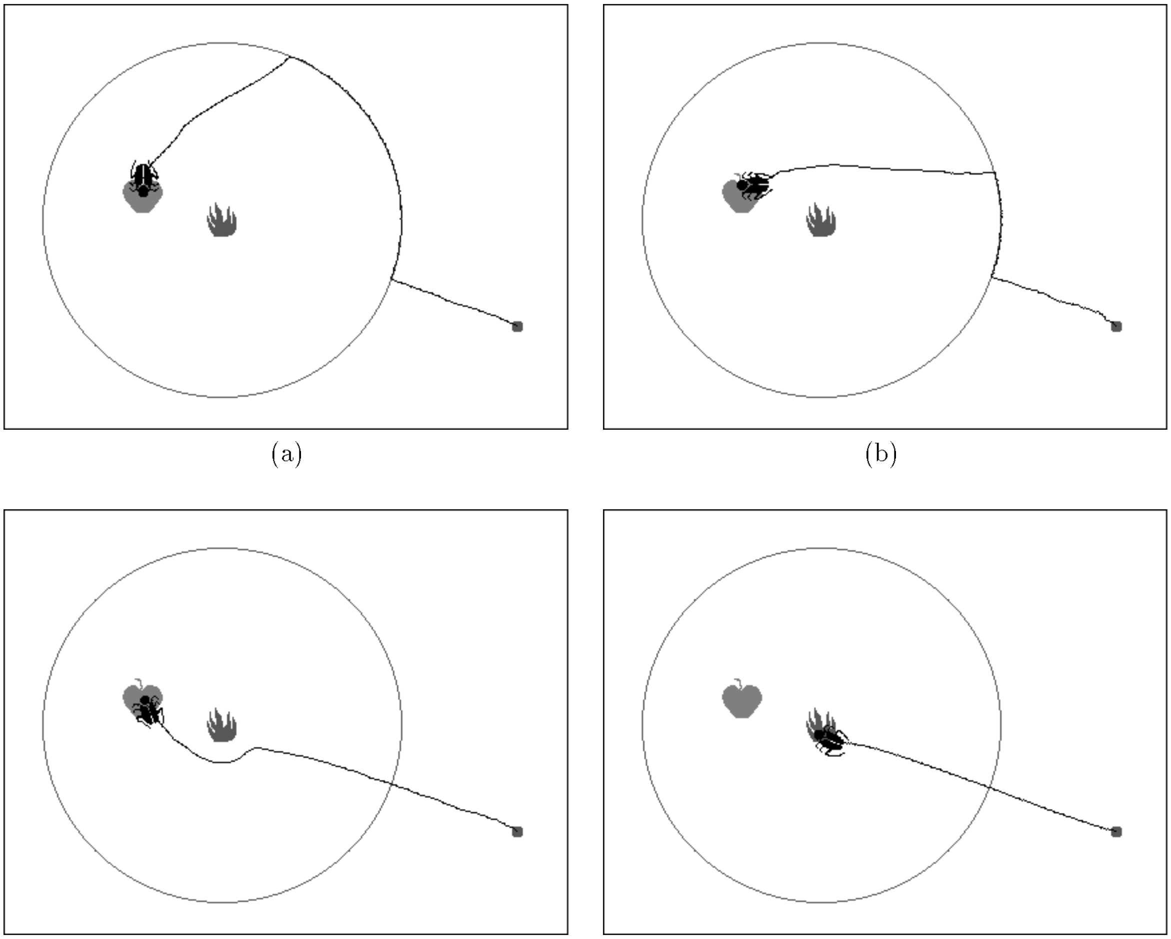


Figure 11: Avoiding danger while searching for food using spatial difference strategy. The creature moves along the boundary of the danger zone to reach the food.

simulation result shows that, despite the simplicity of the creature's network architecture and search strategy, it is able to negotiate obstacles without being taught how to do so.

5.2 Danger Avoidance

Searching for food while avoiding danger can be a complex task. The creature is attracted to appetitive stimulus but repelled by aversive stimulus. It has to find a way to get around the danger in order to locate the food. Figure 11 shows how the creature performs this task using spatial difference method. The food stimulus is spread over a larger region than the danger zone denoted by the circle. The creature is attracted towards the food and repelled by the aversive stimulus whenever it crosses the boundary of the danger zone. In a manner similar to obstacle negotiation, the creature continues its advancement towards the food by moving along the boundary where the repulsion of the aversive stimulus cancels the attraction of the appetitive stiumuls.



(a) (b) (c) (d)

Figure 12: (a) When the creature is not hungry, it takes a large detour to reach the food. (b, c) When it becomes hungry, it takes risks by going nearer to the danger source; the more hungry it is, the nearer it gets to danger. (d) When the creature is very hungry, it becomes suicidal and goes straight into danger.

5.3 Risk Taking

Figure 12(a) shows the experimental setup in which a food happens to fall within the danger zone of an aversive stimulus source. The shortest path to the food intersects the danger source. When the creature is not hungry, it avoids the shortest path and takes a large detour along the boundary to reach the food. However, if the creature starts off with a hungry stomach (i.e., hunger level $H > 1$), it becomes more attracted to the food and takes a route that is closer to the danger source (Fig. 12b). The more hungry is the creature, the more risks it takes, and the closer it gets to the danger source (Fig. 12c). As one would anticipate, when the creature is very hungry, it can become suicidal and can go straight into danger (Fig. 12d).

6 Discussion

The olfactory-motor coordination networks described in this paper have very simple architecture consisting of sensors, maps, and direction vectors. The maps compute temporal or spatial difference of stimulus intensity which biases the creature's heading towards food and away from danger. In addition, detection of obstacle by touch sensors biases the creature away from the obstacle, while random Wandering Vector biases it away from its current heading. As a result of the interactions between the network components, the creature can negotiate obstacle and avoid danger while searching for food (Section 5). It can also identify the dominant stimulus among multiple stimuli present in the environment (Section 3).

To be effective in food searching, the temporal difference network requires an appropriate amount of random wandering (Section 4.1), which in turn causes the creature's search path to become tortuous. As a result, the creature which uses temporal difference strategy takes the largest amount of time to locate the food (Section 4). Nevertheless, when food is abundant, the creature can still find them via more direct routes (Section 3.2).

The paths taken using the other search strategies are more direct. Although the spatial difference network has only two smell sensors and two sets of maps, its performance is comparable to those of the local maximum and gradient ascent networks which have eight sensors and eight sets of maps (Sections 3, 4). In addition, its performance in a noisy environment is as robust as those of the local maximum and gradient ascent networks (Section 4.2).

In comparison, existing works on olfactory-motor coordination focus mainly on the computational modeling of insects' flight paths to a pheromone source (Preiss and Kramer, 1986; Kramer, 1996; Belanger and Willis, 1996; Willis and Arbas, 1996). For example, Belanger and Willis (1996) modeled the structures of laminar and turbulent plumes. They then illustrated by computer simulations that a moth requires appropriate guidance mechanisms to show the characteristic zigzag flight paths along the plume to the source. This paper complements existing works by illustrating the neural mechanisms required to identify dominant stimulus among multiple stimuli and to search for food while negotiating obstacle and avoiding danger. It also compares the performance of a series of increasingly complex network architectures, and shows that there is no significant advantage for a creature to have more than two smell sensors.

Despite the simplicity of the network architecture, complex behaviors emerge from the interactions between various network components and between the creature and the environment. The creature is not a passive object that just processes input stimuli. Instead, it is an active entity that also acts in the environment. The creature performs a *perception-action cycle* (Arbib, 1989): it senses the stimuli, computes intensity differences, decides the next heading, takes a step in that direction, and the cycle repeats. This approach has been widely adopted in the study of animal behaviors, such as toad's prey-catching, detour, and predator-avoidance behaviors (Arbib, 1987; Liaw and Arbib, 1993; Corbacho and Arbib, 1995).

It is most interesting to compare Arbib's model of a computational toad (Arbib, 1987; Arbib and House, 1987) with the networks described in this paper. The computational toad contains a visual processing module, called *visual schema*, for detecting worm (food) and another for detecting barrier (obstacle). Separate visual depth maps for food and obstacle are maintained in the two schemas. The results produced by the visual schemas are processed by several motor schemas that perform various actions such as snapping, sidesteping, orienting, and jumping. The networks presented in this paper also maintains two sets of maps, one for detecting food and the other for detecting danger. The results computed by these maps are combined with the direction vectors to determine the new heading. Therefore, the food and danger maps can be viewed as neural implementations of olfactory schemas and the direction vectors as part of motor schemas.

The *subsumption architecture* of intelligent robots proposed by Brooks (1991) consists of three layers of control, each building on top of a lower layer. The lowest layer contains components that sense sonar inputs, watch for possible collisions, move the robot forward, turn the robot, etc. The second layer consists of components that perform wandering and collision avoidance. The highest layer contains modules that look for an interesting place to explore and plan a path towards the destination. Each of these components can be interpreted as a sensory, motor, or sensorimotor coordination schema. The networks described in this paper provides examples of how the above schemas may be implemented in distributed neural networks. For example, the exploratory schema would correspond to the Food Maps, Food Units, and the Forward Direction Vector. The collison and danger avoidance schema would correspond to the Danger Maps, Danger Units, Backward Direction Vector, and the Touch Reflex Vector. Finally, the wandering schema would correspond to the Wandering Vector. This way, a schema may be implemented by several neural networks distributed over large regions of the creature's brain.

This paper has focused on exploratory behavior in a static environment in which food, danger, and obstacles remain stationary. Preliminary study has also been conducted in a dynamic environment in which the food sources are free to move about. It is found that the creature is able to track down moving food if it moves fast enough. More research on exploration in a dynamic environment needs to be performed. Possible topics include wind effects, creation and depletion of food sources, mapping of the dynamic environment, etc. In the current implementation, the network's connection weights that encode stimulus patterns are pre-coded. It would be useful to study how the weights can be learned and how the learning process affects the creature's behavior, especially in a dynamic world.

7 Conclusion

This paper presented four types of neural networks capable of searching for food, negotiating obstacle, and evading danger. The temporal difference network has only one smell sensor and requires the least amount of neural hardware. It represents a minimal network architecture capable of performing the tasks. Due to the simplicity of the network, the creature tends to take less direct, more tortuous routes towards food and away from danger. Nevertheless, if food is abundant, the creature can still reach them via more direct paths. The spatial difference network has two smell sensors and two sets of food and danger maps. A creature equiped with such a network can perform the tasks by taking more direct paths. The local maximum and gradient ascent networks have eight smell sensors and eights sets of maps. However, these networks do not perform significantly better than the spatial difference network. In order words, there is no advantage to have more than two sensors and two sets of maps. This experimental result may help to explain, from the point of view of computational power, why real animals have only one or two smell sensing organs.

The networks' architectures are very simple. They consist of simple maps for computing temporal or spatial difference of stimulus intensity and direction vectors that bias the creature's heading. There is no complex system for path finding or route planning. Despite the networks' simplicity, complex behaviors such as searching for food while negotiating obstacle and avoiding danger, risk taking, and suicidal behavior emerge from the creature's interaction with the environment. This research therefore illustrates a possible direction for studying more complex behaviors of intelligent beings.

Acknowledgement

I wish to thank Risto Miikkulainen, Ray Mooney, and their students for their valuable feedback when I presented an earlier version of this work to them. I also like to thank my students Hsueh-Chyi Yee and Chung-Yong Lim for implementing the graphical user interface and earlier versions of the programs. This research is supported by NUS Academic Research Fund RP960684.

Appendix

A Computing Dominant Stimulus Intensity

To see how the Food Maps and the Food Units compute the intensity of the dominant stimulus, let us consider the following example. The sensor's activity in response to a single stimulus p is given by

$$a_j = I_p u_{pj} \tag{15}$$

where I_p is the intensity of the stimulus and u_{pj} is a normalized pattern corresponding to type p stimulus. The pattern u_{pj} is identical to w^+_{ij} for a particular unit b^+_i that recognizes the stimulus. In the presence of several stimuli, the sensor's activity become

$$\begin{aligned} a_j &= \sum_p I_p u_{pj} \\ \bar{a}_j &= \frac{\sum_p I_p u_{pj}}{\sum_j \sum_p I_p u_{pj}} \ . \end{aligned} \tag{16}$$

The food map units b_i^+ compute weighted-sums of the sensor's activities:

$$b_i^+ = \frac{\sum_j \sum_p I_p u_{pj} w_{ij}^+}{\sum_j \sum_p I_p u_{pj}} \ . \tag{17}$$

Since u_{pj} and w_{ij}^+ are normalized, the weighted sum $\sum_j u_{pj} w_{ij}^+$ attains the largest value when $w_{ij}^+ = u_{pj}$ for each j. Suppose that stimulus k is the dominant stimulus, i.e., $I_p < I_k$ for $p \neq k$. Then, the largest b_i^+ is the one with $i = k$, which implies that $w_{ij}^+ = u_{kj}$ for each j. As a result of Eq. 4 and 5, $c_k^+ = 1$, $c_i^+ = 0$ for $i \neq k$, and

$$d_j^+ = a_j m_{jk}^+ = \sum_p I_p u_{pj} w_{kj}^+ \ . \tag{18}$$

Assuming that the sensor's activity patterns in response to individual stimuli do not overlap significantly, then $u_{pj} w_{kj}^+ \approx 0$ for $p \neq k$, which leads to the results:

$$d_j^+ \approx I_k {w_{kj}^+}^2 \tag{19}$$

$$e^+(t) = I_k \sum_j {w_{kj}^+}^2 \ . \tag{20}$$

Since w_{kj}^+ are normalized weights, $\sum_j {w_{kj}^+}^2$ is a constant for each type of stimulus. Thus, $e^+(t)$ is a good measure of the intensity I_k of the dominant stimulus.

References

Arbib, M. A. (1987). Levels of modeling of mechanisms of visually guided behavior. *Behavioral and Brain Sciences*, 10:407–465.

Arbib, M. A. (1989). *The Metaphorical Brain 2: Neural Networks and Beyond.* New York: Wiley.

Arbib, M. A., and House, D. H. (1987). Depth and detours: An essay on visually guided behavior. In Arbib, M. A., and Hanson, A. R., editors, *Vision, Brain, and Cooperative Computation.* Cambridge, Massachusetts: MIT Press.

Arbib, M. A., and Lee, H. B. (1993). Anuran visuomotor coordination for detour behavior: From retina to motor schemas. In Meyer, J.-A., Roitblat, H. L., and Wilson, S. W., editors, *From Animals to Animats 2*, 42–51. Cambridge, Massachusetts: MIT Press.

Belanger, J. H., and Willis, M. A. (1996). Adaptive control of ordor-guided locomotion: Behavioral flexibility as an antidote to environmental unpredictability. *Adaptive Behavior*, 4(3/4):217–253.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47:139–159.

Corbacho, F. J., and Arbib, M. A. (1995). Learning to detour. *Adaptive Behavior*, 3(4):419–468.

Grasse, K. L., and Cynader, M. (1991). The accessory optic system in frontal-eyed animals. In Leventhal, A. G., editor, *The Neural Basis of Visual Function.* London: Macmillan Press.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (1991). *Principles of Neural Science.* Norwalk, Connecticut: Appleton & Lange. 3rd edition.

Kramer, E. (1996). A tentative intercausal nexus and its computer model on insect orientation in windborne pheromone plumes. In Cardé, R. T., and Minks, A. K., editors, *Pheromone Research: New Directions.* London: Chapman and Hall.

Liaw, J.-S., and Arbib, M. A. (1993). Neural mechanisms underlying direction-selective avoidance behavior. *Adaptive Behavior*, 1(3):227–261.

Masson, C., and Linster, C. (1995). Towards a cognitive understanding of odor discrimination: Combining experimental and theoretical approaches. *Behavioral Processes.*

Meyer, J.-A., and Guillot, A. (1991). Simulation of adaptive behavior in animals: Review and prospect. In Meyer, J.-A., and Wilson, S. W., editors, *From Animals to Animats 1*, 2–14. Cambridge, Massachusetts: MIT Press.

Mura, F., and Franceschini, N. (1994). Visual control of altitude and speed in a flying agent. In Cliff, D., Husbands, P., Meyer, J.-A., and Wilson, S. W., editors, *From Animals to Animats 3*, 91–99. Cambridge, Massachusetts: MIT Press.

Preiss, R., and Kramer, E. (1986). Mechanism of pheromone orientation in flying moths. *Naturwissenschaften*, 73:555–557.

Webb, B. (1994). Robotic experiments in cricket phonotaxis. In Cliff, D., Husbands, P., Meyer, J.-A., and Wilson, S. W., editors, *From Animals to Animats 3*, 45–54. Cambridge, Massachusetts: MIT Press.

Willis, M. A., and Arbas, E. A. (1996). Adaptive behavior and reflexive responses: Another perspective on odor-guided locomotion. In Cardé, R. T., and Minks, A. K., editors, *Pheromone Research: New Directions.* London: Chapman and Hall.