Novelty and Reinforcement Learning in the Value System of Developmental Robots

Xiao Huang and John Weng
Computer Science and Engineering Department
Michigan State University
East Lansing, MI, 48824

Abstract

The value system of a developmental robot signals the occurrence of salient sensory inputs, modulates the mapping from sensory inputs to action outputs, and evaluates candidate actions. In the work reported here, a low level value system is modeled and implemented. It simulates the non-associative animal learning mechanism known as habituation effect. Reinforcement learning is also integrated with novelty. Experimental results show that the proposed value system works as designed in a study of robot viewing angle selection.

1. Introduction

Motivated by studies of developmental psychology and neuroscience (Piaget, 1952) (Flavell et al., 1993) (Sur et al., 1999), computation studies about autonomous mental development has drawn increased attention (Weng et al., 2000) (Almassy et al., 1998) (Ogmen, 1997). With the developmental paradigm for robots, a task-nonspecific developmental program is designed by human programmer. The robot develops its mental skills through real-time, online interactions with the environment. An important part of a developmental program is its value system.

Neuroscience studies have shown that value system has the basic function of the multiple diffuse ascending systems of the vertebrate brain (Montague et al., 1996) (Sporns, 2000). The detailed mechanisms of the value system and its development are mostly unknown although some characterizations of this system are available (Schultz, 2000). Generally, value systems are distributed in the brain. They respond to sensory stimuli, modulate neural activity, and project the effect to wide areas of the brain.

Value-dependent learning has been successfully applied to modeling the sensory maps in the barn owl's inferior colliculus (Rucci et al., 1997). Sporns and colleagues (Sporns et al., 2000) proposed a value system based on this learning mechanism to model robots' adaptive behavior. Their work shows that a robot's value system can modulate its own responses in the context of various conditioning tasks. Although reinforcement learning for robots is not new and has been widely studied (Watkins, 1992) (Sutton and Barto, 1998), studies on integrated value systems in robots are still few. Ogmen's work (Ogmen, 1997) is very similar to our study. His framework is based on ART (Adaptive Resonance Theory), which considers novelty, reinforcement and habit. However, only a simple simulation experiment is reported. Whether the model can be used in real time and complex environments is unknown.

In this paper, we report the development of a robotic value system by integrating novelty and reinforcement learning. The novelty models the habituation effect in animal learning. It is known that animals respond differently to stimuli of different novelties. Human babies get bored by constant stimuli. This is displayed by a reduction in fixation time (Kaplan et al., 1990). Infants pay longer attention to novel stimulus. However, this doesn't means that novelty is always preferred (Zeaman, 1976). We propose a computational model of a low level value system which integrates novelty and other rewards. We present the working of this value system through simulation and real time testing on our SAIL (short for Self-organizing, Autonomous, Incremental Learner) robot. The work reported here does not model high-level mechanisms such as stress.

2. System architecture

The basic architecture implemented for the SAIL robot is shown in Fig. 1. The sensory input can be visual, auditory, and tactile. These inputs are represented by a high dimensional vector so that each component corresponds to a scale-normalized receptor (e.g. pixel). It is the cognitive mapping module that derives most discriminating features from input streams and maps each input vector to the corresponding effector control signal.

Mathematically, the cognitive mapping is formu-

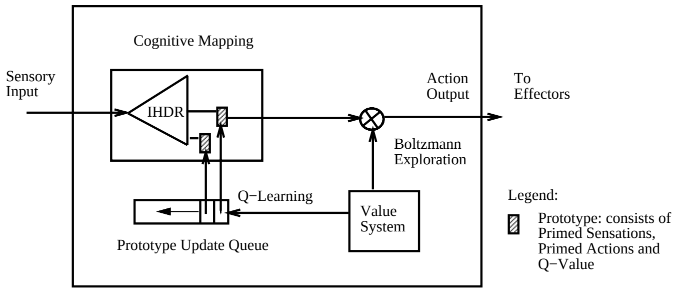


Figure 1: System architecture of SAIL experiments.

lated as a mapping M: $S \times X \rightarrow X' \times A \times Q$, where S is the state (context) space, X the current sensory space, X' the space of primed sensation, Q the space of value and A is the action space. In every time step, M accepts the current sensory input $x(t)$ and combines it with the current state $s(t)$ to generate the primed sensation $X'(t)$ and the corresponding action output $a(t+1)$. The cognitive mapping is realized by the Incremental Hierarchical Discriminant Regression (IHDR) tree (Hwang and Weng, 1999) (Weng and Hwang, 2000), which derives the most discriminating features and uses a tree structure to find the best matching in a fast logarithmic time. Compared with other methods, such as artificial neural network, linear discriminant analysis, and principal component analysis, IHDR has advantages of dealing with high-dimensional input, deriving discriminant features autonomously, learning incrementally, allowing one-instance learning, and low time complexity.

The current context is represented by context vector $c(t) \in S \times X$. Given $c(t)$, the IHDR tree finds the best match prototype c' among a large number of candidates. Each prototype c' is associated with a list of primed contexts: $\{p_1, p_2, ..., p_k\}$. In each primed context, p_i consists of primed sensation x_p, primed action a_p and corresponding Q value. The primed sensation is what the robot predicts to sense after taking the corresponding primed action. The Q value is the expected value of the corresponding action. Given the list of primed contexts, it is the value system that determines which primed action should be taken based on its Q-value. Reinforcement learning is integrated into the value system. After taking one action, the robot enters a new state. The value system calculates the novelty by computing the difference between the current sensation in the new state and the primed sensation in the last state. Then the novelty is combined with immediate reward to update the Q value of the related actions. More detail about the value system is discussed in the next section.

A major problem with a reward is that it is typically delayed. The idea of value backpropagation used in Q-learning is applied here. A challenge of online incremental development is that global iteration is not allowed for speed considerations. We use a first-come first-out queue, called prototype update queue, which stores the addresses of formerly visited primitive prototypes. This queue keeps a history of prototype through which reward is backpropagated recursively but only locally, avoiding updating all the states (which is impossible for real-time development).

3. The value system

The value system of a developmental robot signals the occurrence of salient sensory inputs, modulates the mapping from sensory inputs to action outputs, and evaluates candidate actions (Sporns, 2000) (Montague et al., 1996). The value system of the central nervous system of a robot at its "birth" time is called innate value system. It further develops continuously throughout its "life" experience. The value system of a human adult is very complex. It is affected by a wide array of social and environmental factors. The work reported here deals with only some basic low level mechanisms of the value system, namely, a novelty and reward based integration scheme.

3.1 Spatial and temporal bias

The innate value system of a developmental robot is designed by the programmer. It includes the following two aspects: innate spatial bias and innate temporal bias. The term "spatial" here means different sensory elements with different signal preferences are located at different locations of the robot body. The term "temporal" means that the spatial bias changes with time. For example, a pain signal from a pain sensor is assigned a negative value and a signal from a sweet taste is assigned a positive value. This is an innate spatial bias. If a pain sensor continuously sends signal to the brain for a long time period, a newborn does not feel the pain as strong as it is sensed for the first time. This is a temporal bias (Domjan, 1998).

3.2 Novelty and immediate reward

Novelty plays a very important role in both nonassociative learning and classical conditioning. It is a part of the value measured by the value system.

As shown in Fig. 1, every prototype retrieved from the IHDR tree consists of 3 lists: primed sensations $X = (x_{p1}, x_{p2}, ..., x_{pn})$, primed actions $A = (a_{p1}, a_{p2}, ..., a_{pn})$ and corresponding Q values $Q = (q_{p1}, q_{p2}, ..., q_{pn})$. The innate value system evaluates each action a_{pi} in the primed action list A and each sensation vector x_{pi} in the primed sensation list X. The evaluation integrates novelty and rewards.

The novelty can be measured by the agreement between what is predicted by the robot and what the robot actually senses. If the robot can predict well what will happen, the novelty is low. Then we can define novelty as the normalized distance between the selected primed sensation $x_{pi} = (x'_1, x'_2 ... x'_m)$ and the actual sensation $x(t+1)$ in next time:

$$n(t) = \sqrt{\frac{1}{m} \sum_{j=1}^{m} \frac{(x'_j(t) - x_j(t+1))^2}{\sigma_j^2(t)}} \qquad (1)$$

where m is the dimension of sensory input. Each component is divided by the expected deviation σ_j, which is the time-discounted average of the squared difference $(x'_j - x_j)^2$. Based on IHDR, only when the sensory input is much different from retrieved prototype, will a new prototype be generated.

Suppose that a robot baby is staring at a toy for a while. Gradually, the primed sensation x_p can match the actually sensed sensation well: "I will see that puppy sitting this way next time." Thus the current action, staring without changing, reduces its value in the above expression, since $n(t)$ drops. Then, another action, such as turning away to look at other parts in the scene, has a relatively higher value. Thus, the robot baby turns his eyes away.

It is necessary to note here that the novelty measure $n(t)$ is a low level measure. The system's preference to a sensory input is typically not just a simple function of $n(t)$. Besides novelty, human trainer and environment can shape the robot's behaviors through its biased sensors. A biased sensor is one whose signal has an innate preference pattern by the robot. For example, a biased sensor value $r = 1$ if the human teacher presses its "good" button and $p = -1$ if the human teacher presses its "bad" button. Now, we can integrate novelty and immediate reward so that the robot can take both factors into account. The combined reward is defined as a weighted sum of physical reward and the novelty:

$$r(t) = \alpha p(t) + \beta r(t) + (1 - \alpha - \beta) n(t) \qquad (2)$$

where $0 < \alpha, \beta < 1$ is an adjustable parameter indicating the relative weight between $p(t)$, $r(t)$ and $n(t)$, which specify punishment, positive reward and novelty. researches in animal learning show that different reinforcers has different effect. Punishment typically produces a change in behavior much more rapidly than other forms of reinforcers (Domjan, 1998). So in our experiments, $\alpha > \beta > 1 - \alpha - \beta$.

We have, however, two major problems. Firstly, the reward r is not always consistent. Human may make mistakes in giving rewards, and thus, the relationship between an action and the actual reward is not always certain. The second is the delayed reward problem. The reward due to an action is typically delayed since the effect of an action is typically not known until some time after the action is complete. These two problems are dealt with by the following Q-learning algorithm.

3.3 Q learning algorithm and Boltzmann exploration

Q-learning is one of the most popular reinforcement learning algorithm (Watkins, 1992). The basic idea is as follows. Keep a Q value for every possible pair of primed sensation x_p and every possible action a_p: $Q(x_p, a_p)$, which indicates the value of action a_p at current state s. The action with the largest value will be selected as output and then a reward $r(t+1)$ will be received. The Q-learning updating expression is as follows:

$$\begin{aligned} Q(x_p(t), a_p(t)) &:= (1-\alpha) Q(x_p(t), a_p(t)) \\ &+ \alpha(r(t+1) + \gamma \max\nolimits_{a'} Q(x_p(t+1), a_p(t+1))) \end{aligned} \qquad (3)$$

where α and γ are two positive numbers between 0 and 1. The parameter α is the updating rate. The larger it is, the faster the Q value is updated by the recent rewards. The parameter γ is for discount in time. With this algorithm, Q-values are updated according to the immediate reward $r(t+1)$ and the value of the next sensation-action pair, thus delayed reward can be back-propagated in time during learning. Because lower animals and infants only have developed a relatively simple value system, they should be given rewards immediately after a good behavior whenever possible. This is a technique for successful animal training.

Early estimated Q value should not be overtrusted, since they are not good before other actions are tired. We applied Boltzmann exploration to Q-learning algorithm (Sutton and Barto, 1998). At each state (primitive prototype) the robot has a list of action $A(S) = (a_{p1}, a_{p2}, ..., a_{pn})$ to choose from. The probability for action a to be chosen at s is:

$$p(s, a) = \frac{e^{\frac{Q(s,a)}{\theta}}}{\Sigma_{a' \in A(s)} e^{\frac{Q(s,a)}{\theta}}} \qquad (4)$$

where θ is a positive parameter called temperature. With a high temperature, all actions in $A(s)$ almost have the same probability to be chosen. When $\theta \to 0$, Boltzmann exploration more likely chooses action a that has a high Q value. With this exploration mechanism, actions with smaller Q value are still possible to be chosen so that action space can be explored. Another effect of Boltzmann exploration is to avoid local minima, like always paying attention to certain part and not being able to notice novel thing in other views.

3.4 Prototype updating queue

In the batch learning mode of a Q-learning algorithm, the back-propagation is applied to all states. For real-time development, this global iteration method is not applicable, due to the excessive time required. We must use a local method that only involves a small number of computations that go through a local state trajectory. This is why we designed the prototype updating queue in Fig. 1, which stores the addresses of formerly visited primitive prototypes. At each time step, after the sensory input is received, the primed sensation is updated according to the following expression:

$$x^{(n)}(t) := x^{(n-1)}(t) + \frac{1+l}{n}\gamma(x(t+1) - x^{(n-1)}(t)) \quad (5)$$

where l is the amnesic parameter. If $l > 1$, it means the latest input contributes more.

Thus, not only is the Q value backpropagated, so is the primed sensation. This back propagation is performed recursively from the tail of the queue back to the head of the queue. After the entire queue is updated, the current primitive prototype's address is pushed into the queue and the oldest primitive prototype at the head is pushed out of the queue. Because we can limit the length of prototype queue, real-time updating becomes possible.

3.5 Algorithm of innate value system

The algorithm of the innate value system works in the following way:

1. Grab the new sensory input $x(t)$.

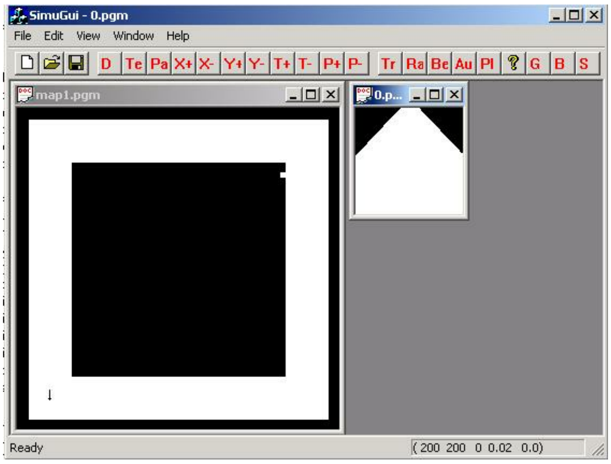
2. Query the IHDR tree and get a prototype $s(t)$ and related list of primed contexts.

3. If $x(t)$ is significantly different from $s(t)$, it is considered as a new prototype and we update IHDR tree by saving $x(t)$. Otherwise, $x(t)$ updates $s(t)$ through incremental averaging.

4. Using Boltzmann Exploration Eq. 4 to chose an action based on the Q-value of every primed action. Execute the action.

5. Calculate novelty with Eq. 1 and integrate with immediate reward $r(t+1)$.

6. Update prototype queue with Eq. 3 and Eq. 5. Go to step 1.



Ready

(200 200 0 0.02 0.0)

Figure 2: The GUI simulator. The arrow indicates the position and the viewing angle of the robot.

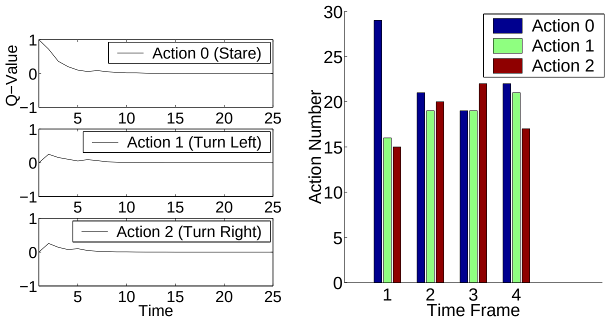
4. Simulations

In order to test the innate value system with ground truth, a simulation environment is developed. The simulator GUI is shown in Fig. 2. The big window shows the viewing environment while the small window shows the images the robot observes currently. There are several buttons that control the position and viewing angle of the robot. The "Good" and "Bad" buttons are used to issue rewards. In every state, the baby robot has three possible actions: stay at the current viewing angel (action 0), turn neck left 30 degree (action 1) and turn neck right 30 degree (action 2). The representation of sensory input consists of visual images and absolute viewing angle. The dimension of input image is 100 × 100. We assume that the robot cannot look backward and the number of absolute viewing angle is 7 (from -3 to 3, 0 stands for center). The parameters are defined as follows: $\alpha = 0.8, \gamma = 0.9$ in Eq. 3; the initial value of θ is 10 in Eq. 4.

4.1 Habituation effect

In the first experiment, we let the robot explores by itself by viewing around. It is reasonable that a positive initial Q-value (e.g. 1) is assigned to action 0, which assumes that the robot just stares staticly. Only when one view is really boring, it will turn its head away. The initial Q-value of other actions is 0. Fig. 3 shows how the Q-value of each action changes based on novelty in the state whose absolute view angle is 0. As shown in the left part, for action 0, it starts with a positive Q-value, which means the probability of staying at the same viewing angle is large. After staring for a while, the primed sensation of action 0 is equal to the actual sensation of next step. According to Eq. 1, the novelty value is equal to zero so that the Q-value of primed action

0 decreases. For action 1 and action 2, at the beginning, the primed sensation is set as a long vector in which every element is zero. After taking an action, the current sensation is very different from the initial sensation. That is, the novelty value is high. We can see from Fig. 3 that at first several steps, the Q-values of these two actions increase. However, after we update the primed sensation, the primed sensation will be the same as the actual sensation if the robot takes the action again. Then the novelty becomes zero and Q-value decreases. After a long time training (300 steps), the robot can predict the actual sensation of next step whatever action it takes. So the Q-value of each action converges to the same value (0). This means each action has the same probability to be chosen. The right part of Fig. 3 shows the number of each action in different time frames (60 steps in each time frame). At the beginning, action 0 has a larger Q-value, according to Boltzmann exploration, it has more chance to be chosen. The probability of action 1 and action 2 is almost the same. After 300 steps, the Q-value of each action is nearly equal, so the numbers of each action are close. The experiment shows that because of habituation effect, the robot loses the interest of any action after exploration and just chooses an action randomly.



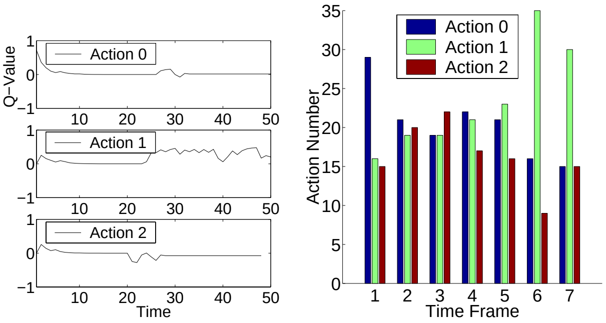
Q-value vs time

Action number

Figure 3: Habituation effect. In the left part: the 1st, 2nd and 3rd plots correspond to the Q-value of action 0, action 1, and action 2 respectively. In the right part, the frequency of actions in different time frames.

4.2 *Integration of novelty and immediate reward*

After the above experiment, we began to issue rewards. For example, when the robot turns left, human teachers give it a positive reward (1). For other two actions, negative rewards (-1) will be issued. Then the actual reward the robot receives is an integration of novelty and immediate reward. For action 0 and action 2, the Q-values change in the same way as in experiment 1, converging to 0. The Q-value of action 1 is always positive because we keep issuing positive rewards. As we can see in the left part of Fig. 4, at the beginning, the Q-value of action 0 is the largest and the robot takes the action with a high probability. After training, the Q-value of action 1 is much larger than that of other actions. As shown in Fig. 4, gradually, action 1 is chosen the most often.



Q-value vs time

Action number

Figure 4: Integration of novelty and immediate reward.

4.3 *Increase novelty with a moving object*

In order to show novelty preference, a moving toy is added to the simulation environment after experiment in Fig. 3. The testing image is shown in Fig. 5. Every time when the robot is in the state with the absolute viewing angle 0, one of these images is generated randomly. Thus, the primed sensation of action 0 is always different from the actual sensation. As shown in left part of Fig. 6, the Q-value of action 0 is positive because of high novelty. In contrast, the Q-values of action 1 and action 2 are more near to zero. After training, the robot found that staying with viewing angel of east is the most interesting. So the action 0 is chosen the most often.

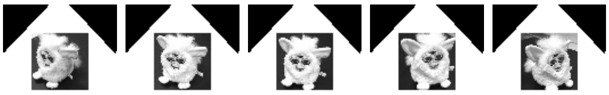
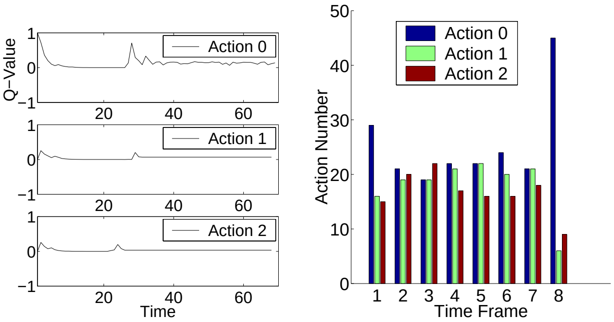


Figure 5: Simulation of a moving object.

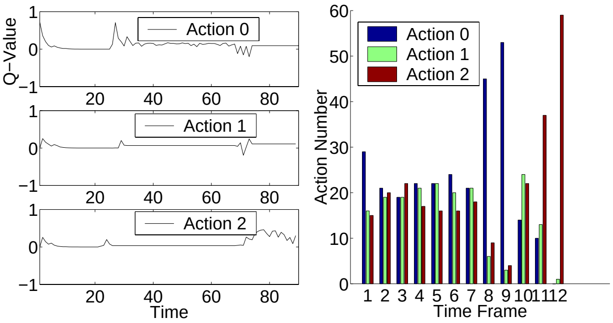
4.4 *Suppress novelty with immediate rewards*

After the third experiment , we issued positive rewards to action 2 (turn right), and negative rewards to action 0. Thus, even though the novelty is high when the robot stares at a moving object, the immediate rewards suppress the novelty. Gradually, the Q value of action 2 increases. As shown in Fig. 7, after training, the robot almost chooses only action 2.



Q-value vs time   Action Number

Figure 6: Increase novelty with a moving object.



Q-value vs time   Action Number

Figure 7: Suppress novelty with immediate rewards.

5. Experiments with SAIL robot

The next experiment is developing value system of our SAIL robot (short for Self-organizing, Autonomous, Incremental Learner). SAIL shown in Fig. 8 is a human-size robot custom-made at Michigan State University. It has two "eyes", which are controlled by fast pan-tilt heads. 28 touch sensors are installed on its arm, neck, head, and bumper to allow human to teach how to act by direct touch. Its drive-base enables it to operate both indoor and outdoor. A high-end dual-processor dual-bus PC workstation with 512 MB RAM makes real-time learning possible. In our real time testing, each step SAIL has 3 action choices: turn its neck left, turn its neck right and stay. Totally, there are 7 absolute positions of its neck. Center is position 0, and from left to right is position -3 to 3 Because there are a lot of noise in real time testing (people come in and come out), we restricted the number of states to be less then 50. The dimension of input image is $30 \times 40 \times 3 \times 2$, where 3 arises from RGB colors and 2 for 2 eyes. The input representation consists of visual images and the absolute position of the robot's neck. The two components are normalized so that each has an equal weight in the representation. The parameters are defined as follows: $\alpha = 0.9, \gamma = 0.9$ in Eq. 3; the initial value of θ is 10 in Eq. 4.

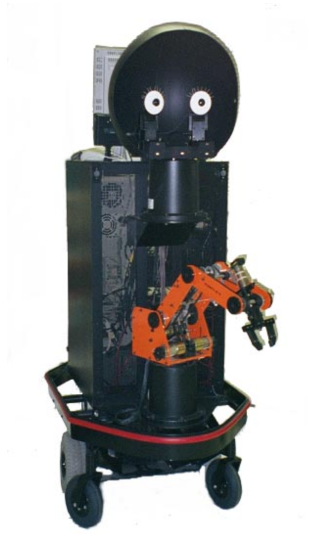


Figure 8: SAIL robot at Michigan State University.

5.1 *Habituation with positive reward*

In order to show the effect of novelty, we let the robot explore by itself for about 5 minutes, then kept moving a toy on the right side of the robot. The absolute neck position is -2. As shown in Fig. 9, the first plot is the Q-value of each action, the second plot is the reward of corresponding action, the third plot is novelty value and the last one is the integrated reward. After exploration (200 steps later), a moving toy increases the novelty of action 0 (stay). At the same time, positive rewards are issued to action 0, so its corresponding Q-value (red line) converges to 1 while the Q-vaules of other two actions converge to 0. The robot kept looking at the toy for about 20 steps. Then we moved the toy to left side (absolute neck position is -1), the novelty of action 1 (turn left) increases (blue line). Finally, at most time, the robot would take action 1. However, at step 420, action 0 is taken again. That is because Boltzmann exploration is applied. After training, the robot would prefer to the Mickey mouse if positive rewards are issued when staring at the toy (Fig. 10.)

The left part of Fig. 11 shows the number of prototypes in each level of the IHDR tree. About 50 prototypes are generated through incremental learning. The depth of the tree is 4. The right part of Fig. 11 shows the computing time of each step in the real time testing. The reason for the changes in time is that the retrieving time for IHDR tree is not exactly constant. For example, if a leaf node keeps more prototypes, its retrieving time increases. The average retrieving time is about 40 ms.

The tree structure is shown in Fig. 12. In the root node ($q = 5$), the first line shows the 5 prototypes, the second line shows the disciminating features represented as images.

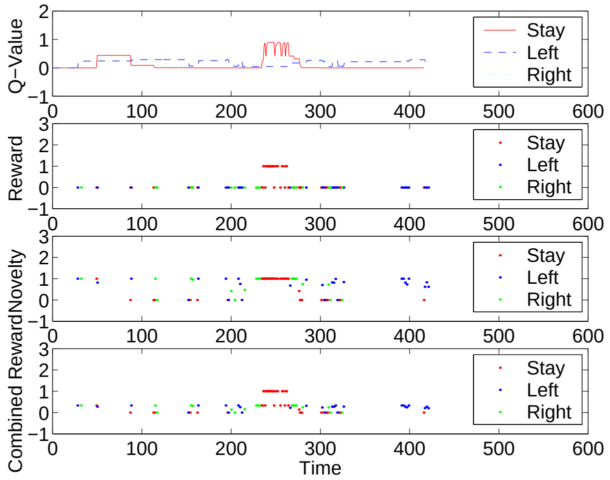


Figure 9: The Q-value, reward, novelty and integrated reward of each action at position -2.

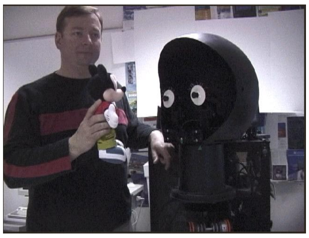
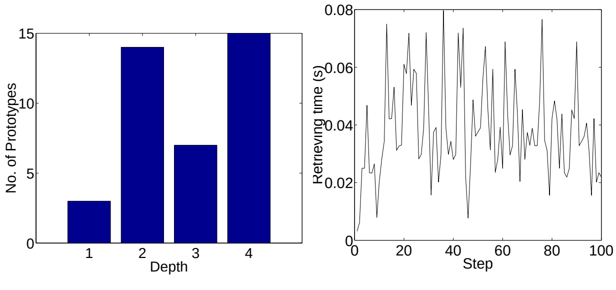


Figure 10: Preference to certain visual stimuli.



No. of prototypes in each level Retrieving time

Figure 11: Real time testing information.

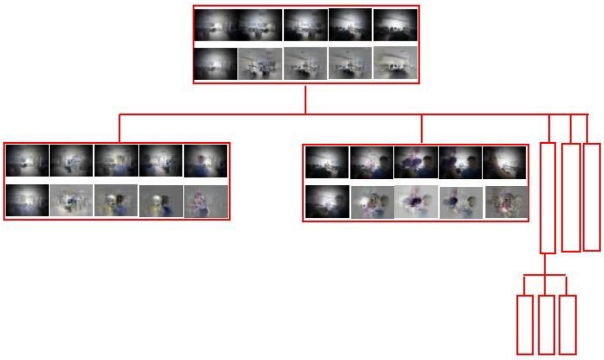


Figure 12: Tree Structure. Each block indicates a tree node. The first row of each node shows the x-cluster centers presented as images. The first image of the second row is the grand mean of all the x-clusters. The remaining images of the second row are the discriminating features represented as images. Here a Gaussian filter is used to alleviate noise.

5.2 *Multiple rewards for different actions*

In this experiment, we gave different rewards to each action at position 2. In the beginning (first 200 steps), we kept moving a toy, so the Q-value of action 0 (stay) is the highest one (the first plot in Fig. 13). The value of novelty is shown in the third plot. Then punishment was issued to action 0 at step 205. Its Q-value became negative. Positive rewards were issued to action 1 and 2 (the second plot). Action 1 got more positive rewards, finally its Q-value became the largest. The fifth plot shows the changes of learning rate. The initial learning rate is 0.9. If rewards are issued, the learning rate decreases (around 0.3), which means that the robot would remember rewards much longer than novelty.

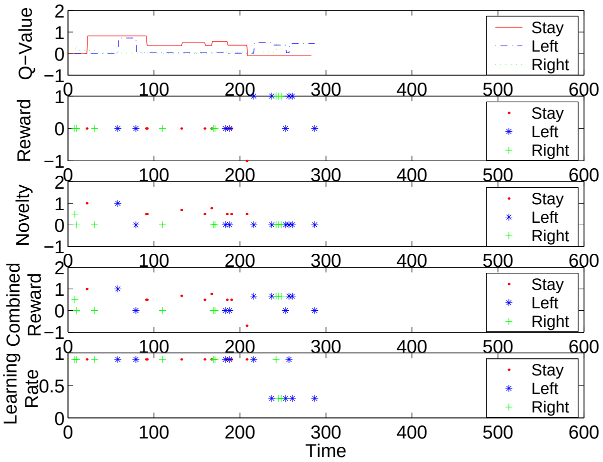


Figure 13: The Q-value, reward, novelty, integrated reward and learning rate of each action at position 2 when multiple rewards are issued.

6. Conclusions

In this paper, we developed a low level value system for a developmental robot. Both simulation and real-time experiments are reported. The value system integrates the habituation mechanism and reinforcement learning. We successfully applied the system to simulate visual attention effect. Our SAIL robot learns to pay attention to salient visual stimuli while neglecting unimportant input. Motivated by psychology studies in instrumental conditioning, we integrated reinforcement learning with habituation so that the robot's responses to certain visual stimuli would change after interacting with human trainers, that is, cognitive development of the robot takes place. Even though the low level value system modeled some adaptive behaviors in animal learning, what we accomplished is still one step towards the challenging autonomous mental development. Our next step is to implement the SHM (Stagger Hierarchical Mapping) (Zhang et al., 2001) method to do local analysis and apply the framework to vision-based outdoor navigation.

Acknowledgments

The authors would like to thank Wey S. Hwang for his major contribution to an earlier version of the IHDR program and Yilu Zhang for his contribution to the Q-learning algorithm. The work is supported in part by National Science Foundation under grant No. IIS 9815191, DARPA ETO under contract No. DAAN02-98-C-4025, and DARPA ITO under grant No. DABT63-99-1-0014.

References

Almassy, N., Edelman, G., and Sporns, O. (1998). Behavioral constraints in the development of neural properties: A cortical model embedded in a real-world device. *Cerebral Cortex*, 8(4):346–361.

Domjan, M. (1998). *The Principles of learning and behavior.* Brooks/Cole Publishing Company, Belmont, CA.

Flavell, J., Miller, P., and Miller, S. (1993). *Cognitive Development.* Prentice-Hall, Englewood Cliffs, NJ.

Hwang, W. and Weng, J. (1999). Hierarchical discriminat regression. *IEEE Trans. on Patten Analysis and Machine Intelligence*, 22(11):1277–1293.

Kaplan, P., Werner, J., and Rudy, J. (1990). Habitutation, sensitization and infant visual attention. In Rovee-Collier, C. and Lipsit, L., (Eds.), *Advances in Infancy Research*, pages 61–110. ABLEX Publishing Corporation, Norwood, NJ.

Montague, P., Dayan, P., and Sejnowski, T. (1996). A framework for mesencephalic dopamine systems based on predictive hebbian learning. *The journal of Neuroscience*, 16(5):1936–1947.

Ogmen, H. (1997). A developmental perspective to neural models of intelligence and learning. In Levine, D. and Elsberry, R., (Eds.), *Optimality in Biological and Artificial Networks?*, pages 363–395. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.

Piaget, J. (1952). *The Origins of Intelligence in Children.* International Universities Press, INC, New York.

Rucci, M., Tononi, G., and Edelman, G. (1997). Registration of neural maps through value-dependent learning: Modeling the alighment of auditory and visual maps in the barn owl's pitic tecturm. *Journal of Neuroscience*, 17:334–352.

Schultz, W. (2000). Multiple reward signals in the brain. *Nature Reviews: Neuroscience*, 1:199–207.

Sporns, O. (2000). Modeling development and learning in autonomous devices. In *Workshop on Development and Learning*, pages 88–94, E. Lansing, Michigan, USA.

Sporns, O., Almassy, N., and Edelman, G. (2000). Plasticity in value system and its role in adaptive behavior. *Adaptive Behavior.*

Sur, M., Angelucci, A., and Sharm, J. (1999). Rewiring cortex: The role of patterned activity in development and plasticity of neocortical circuits. *Journal of Neurobiology*, 41:33–43.

Sutton, R. S. and Barto, A. (1998). *Reinforcement Learning – An Introduction.* The MIT Press, Chambridge, MA.

Watkins, C. (1992). Q–learning. *Machine Learning*, 8:279–292.

Weng, J. and Hwang, W. (2000). An incremental learning algorithm with automatically derived discriminating features. In *Proc. of Fourth Asian Conference on Computer Vision*, pages 426–431, Taipei, Taiwan.

Weng, J., McClelland, J., Pentland, A., Sporns, O., StockMan, I., Sur, M., and Thelen, E. (2000). Autonomous mental development by robots and animals. *Science*, 291:599–600.

Zeaman, D. (1976). The ubiquity of novelty-familiarity effects. In Tighe, T. and Leaton, R., (Eds.), *Habituation: Perspective from child development, Animal Behavior and Neurophysiology*, pages 297–320. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.

Zhang, N., Weng, J., and Huang, X. (2001). Visual-based navigation with a stagger hierarchical mapping. In *Proc. of SPIE Symposium on Intelligent Systems and Advanced Manufacturing*, Boston, MA USA.