International Journal of Humanoid Robotics
Vol. 4, No. 2 (2007) 407–433

World Scientific
www.worldscientific.com

INHERENT VALUE SYSTEMS FOR AUTONOMOUS MENTAL DEVELOPMENT

XIAO HUANG* and JUYANG WENG†

Embodied Intelligence Lab,
Computer Science and Engineering Department, Michigan State University,
East Lansing, MI, 48824 USA
**huangxi4@cse.msu.edu*
†*weng@cse.msu.edu*

Received 22 September 2006
Revised 26 September 2006

The inherent value system of a developmental agent enables autonomous mental development to take place right after the agent's "birth." Biologically, it is not clear what basic components constitute a value system. In the computational model introduced here, we propose that inherent value systems should have at least three basic components: punishment, reward and novelty with decreasing weights from the first component to the last. Punishments and rewards are temporally sparse but novelty is temporally dense. We present a biologically inspired computational architecture that guides development of sensorimotor skills through real-time interactions with the environments, driven by an inborn value system. The inherent value system has been successfully tested on an artificial agent in a simulation environment and a robot in the real world.

Keywords: Developmental learning; value system; habituation; reinforcement learning; visual attention.

1. Introduction

Many studies in developmental psychology and neuroscience have shown that mental development is not governed by a task specific goal. Instead, mental development is driven by innate motivational system (e.g. pleasure seeking, pain avoidance, and novelty seeking) and the motivational system is also developed through experience (e.g. working hard while being tired).[9,22,31] In the field of robots, various studies have been carried out to model mental development computationally.[1–3,16,18,25,40] Our long-term goal is to build a task non-specific developmental paradigm so that a robot can develop its cognitive skills through real-time, online interactions with the environment. By task non-specific, we mean that the tasks that a robot will perform in its lifetime are not fully predictable during the programing time for the developmental program. The task-specific experiences enable acquisition of task-specific skills. In this paper, we focus on one of the major differences between the developmental learning paradigm and the traditional learning paradigm: the inherent

value (or motivational) system. Such an inherent value system needs to be hand-programed into the developmental program, which is to be run by a developmental robot at its birth time.

1.1. *The value system of a developmental robot*

The inherent value system is the initial drive of a developmental robot. Through interactions with the environment, the inherent value system enables a robot to gradually develop a more complex value system. A developmental robot is very different from a traditional robot designed to perform a specific given task because development requires the robot to learn sensorimotor skills that will be shared by an open number of settings. In other words, the same value system has to guide a robot in different environments for different tasks (navigating, understanding audio signals, classifying human faces, etc.). Neuroscience studies have shown that a value system has the basic function of the multiple diffuse ascending systems of the vertebrate brain.[28] The detailed mechanisms of the value system and its development are mostly unknown, although some qualitative characterizations of this system are available.[27] Generally, value systems are distributed in the brain. They respond to salient sensory stimuli, modulate neural activity, and project the effect to wide areas of the brain. In this paper, we propose a framework to model the inherent value system that a robot has programed in before "birth."

The challenges of designing an inherent value system for a developmental robot include: (i) The value system must be applicable to all the possible sensorimotor experiences of various settings. For example, it is not always effective to use only a fixed set of salient features for all the tasks, because salient features for one setting (e.g. motion in intruder detection) may not be so for another task (e.g. driving). (ii) It adapts to different maturation stages. For instance, playing with toys is interesting for youngsters but not much so for adults. (iii) It must provide drive in a temporally dense fashion, because reinforcers (e.g. sweet tastes and pain senses) are not presented very often in a typically living experience. A developmental robot mostly lives during a time where a search for appetitive reinforcers (e.g. sweet tastes or immediate reward) is not a goal (e.g. while a child plays). (iv) The value system must work with a real-time system that incrementally grows and updates representation and memory through interactions with an open, complex real physical world. Without this, the value system is unable to deal with increasingly complex settings in unpredictable environments.

1.2. *Background*

Our work is motivated by neuroscience and animal learning. Before getting into technical details, we need to define some terms.

In animal learning community, decrease in responsiveness produced by repeated stimulation is called habituation.[8,21] This fundamental mechanism of adaptive behavior is found in many animals like *Aplysia*,[6] cats[34] and humans.[7] Even though

the habituation effect is a simple form of adaptation mechanism, it enables animals to pay attention to salient stimuli and neglect familiar stimuli, which is a basic function of a value system. In this work, we model habituation as novelty seeking behavior guided by the value system.

Besides psychological analysis of habituation, several computational models have been reported. Stanley's model[30] uses long-term memory so that an animal habituates more quickly to a stimulus if it has habituated previously. The work of Marsland *et al.*[17] combined habituation and self-organization mapping for a robot to detect novelty. Thrun and Schimidhuber[26,35] also provided implementations of novelty. However, those works are not value systems that can guide an agent's complex behaviors such as operant conditioning (called reinforcement learning in the machine learning community).

Modulating the mapping from sensory inputs to action outputs and evaluating the value of candidate actions are other basic functions of the value system. Reinforcement learning is a general model for adaptive behaviors. Sutton and Barto derived the reinforcement learning theory from animal classical conditioning,[32] which is based on expectation and prediction. More elaborate reinforcement learning models are TD(λ) and Q-learning. The basic idea is to learn what to do — how to map situations to actions — by maximizing a numerical reward signal through a trial-and-error procedure.[33,36]

Although reinforcement learning for robots is not new and has been widely studied, studies on integrating novelty and reinforcement in a general value system are still few in number.

1.3. *Related work*

A few attempts to model inherent value/motivation system have been made in the last decade. Barto[4] has reviewed intrinsic motivational systems in the domain of psychology, neuroscience, machine learning and developmental robotics communities. Value-dependent learning has been successfully applied to modeling the sensory maps in the barn owl's inferior colliculus.[23] The value has been modeled in these fields as punishments (negative values) and rewards (positive values). Sporns and his colleagues[1,29] proposed learning mechanisms to learn more complex behaviors using punishments (bitter) and rewards (sweet) from the environment. Ogmen's work[18] was based on ART (Adaptive Resonance Theory), which took into account not only punishments and rewards, but also the novelty in expected punishments and rewards, where punishments, rewards, and novelty are all based on a single value. Kakade and Dayan[12,13] proposed a dopamine model, which uses novelty and shaping to drive exploration in reinforcement learning, although they did not provide the source of information for novelty nor a computational model to measure the novelty.

This paper is the archival journal version of the earlier more preliminary work we presented in 2002,[10] where we modeled the value system consisting of three

components: punishment, reward, and novelty. In our proposed architecture, novelty is not based on a single value of punishments and rewards, but rather the failure of high-dimensional sensory prediction in expected accuracy that is accumulated from experience (with single-value punishment and reward as a special case). This three-component model enabled the agent to perform not only instrumental conditioning as in Refs. 1, 4, 12 and 13, but also nonassociative learning such as habituation based on high-dimensional (5000-D and above) sensory space. After we presented this work in 2002, Oudeyer and his co-workers proposed a mechanism called Intelligent Adaptive Curiosity (IAC) in 2004 and 2005.[19,20] They showed that this intrinsic motivational system helps a robot (an AIBO dog) to maximize its learning. Their work demonstrated how IAC guided a robot to "focus on situations that are neither too predictable nor too unpredictable" in a seven-dimensional feature space.

Our intention of modeling only these three components (i.e. punishment, reward and novelty) in an inherent motivational system is based on our hypothesis that these three components are sufficient to develop behaviors such as concentrating on important subjects, either through explicit environmental communications (e.g. told by robot trainers) or through indirect linking to the three-component inherent motivational system (e.g. working hard in learning complex subjects will lead to higher pays that better satisfy the three-component inherent value system). For example, whether subjects that are neither too predictable nor too unpredictable are important depends to whether the environment values them. Also, subjects that are hardly predictable can be important (e.g. research). However, our above hypothesis has not yet been verified by the field of developmental robotics and is an interesting subject for future research. Furthermore, a suitable trade-off between a simple but sufficient innate value system and a very complex innate value system in facilitating developmental learning is also an important future research topic.

At the current stage, the field of reinforcement learning is very active, although few studies generalize it to a more general issue of motivational system. It is true that most value/motivation systems use machine learning techniques (e.g. Q-learning to systematically deal with delayed reward issue). However, although these techniques are powerful when dealing with practical learning problems, their links to the biological brain are still coarse, mostly not at the cell level. On the other hand, more biologically linked modeling work also faces the challenge of scaling up to uncontrolled environments.

1.4. *Novelties and importance of this work*

Although fully demonstrating scaling-up capabilities of value system development requires more studies and longer developmental time than is possible here, the reported value system proposes the following novel ideas:

(i) Our work reported here is the first implemented inherent value system that integrates three types of values: punishment, reward and novelty.

(ii) We introduce primed sensation (what is predicted by the robot) as a prediction mechanism to enable the value system to further develop through experience, i.e. to predict beyond the immediate value (punishment, reward and novelty).
(iii) The value system is integrated with the system architecture aiming to deal with the four challenges conjunctively listed in Sec. 1.1. This is the first value system capable of dealing with these challenges conjunctively. In other words, the value system can guide a robot to learn online in the real world.

Computational studies of reinforcement often model rewards into a single value delivered from a separate reward channel.[32] This single-value modeling facilitates understanding and simplifies computation. However, primed sensation has been neglected. The value of an action under a state is determined by the rich nature of the primed sensation, not just a global value. For example, hunger and stomach upset are both aversive stimuli: the nature of these different events is needed to adopt actions — eat food for the former and do not eat certain food in the latter. In the value system presented here, the value is not only derived from separate reward channels but also from primed sensation. Furthermore, reinforcers are typically sparse in time: they are delivered at infrequent spots along the time axis. Novelty from primed sensation is however dense in time, defined at every sensory refresh cycle. Thus, a novelty-based value system enables continuous exploration of a developmental robot. Also, in this way, a developmental robot can develop through multiple value channels (novelty, reward and punishment).

Why do we say that novelty in high-dimensional space poses a very challenging problem? For example, if the visual input is a 30×40-pixel color image, the dimension of primed/actual sensation is $30 \times 40 \times 3 = 3{,}600$, where "3" represents RGB components of each pixel. In other words, the dimension of the input space (and also its internal state) is 3,600. This is very different from modeling of novelty in expected single-value reward. It is a very challenging problem and is addressed by our architecture and the Incremental Hierarchical Discriminant Regression (IHDR)[39] method, simulating cortical mapping (details are in Sec. 2).

This value system can also fulfill the challenges listed in Sec. 1.1 through the designed architecture and the technique we used. Because we do not use task-specific features but features developed from experiences, it is open to complex and unknown environments; IHDR can deal with high-dimensional input by accepting raw images (after intensity normalization) directly. It provides online, incremental learning capability, critical for open-ended development. Due to the automatic development of discriminant features and the use of a tree structure, it is efficient to run in real-time even when the memory is large, and can learn and adapt to new environments while performing (i.e. there is no need to have two separate modes for learning and performance[38]). With our proposed value system, it is possible for a robot to develop in new environments that are unknown during the programing time.

To demonstrate how the value system works, we chose a challenging behavior domain with a high-dimensional sensory input: visual attention through neck pan

actions. 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.[14] Infants pay longer attention to novel stimuli. Visual attention has been investigated by many computer vision researchers.[11,15] Important salient features for one setting are not necessarily important ones for another setting. Our approach is fundamentally different from these traditional approaches in that we cast visual attention selection as sensorimotor behaviors developed incrementally from interactions with the environment, driven by the inherent value system as well as the current developed value system. For example, our value system does not define fixed saliency of features, but instead novelty based on experience. A novel stimulus for one robot at one time is not novel if it is sensed repeatedly for long by the same robot. Furthermore, our experiments were conducted in the real world.

In what follows, we first review the architecture of the system. The detailed value system is presented in Sec. 3. The simulation and real-time experimental results are reported in Secs. 4 and 5, respectively. Then, we discuss the limitation of current work in Sec. 6 and draw our conclusions in Sec. 7.

2. System Architecture

The basic architecture of developmental learning is shown in Fig. 1. Before explaining this architecture, we need to give some definitions.

Observation driven state transition: we define context $l(t)$ as any information related to the agent at time t. The developmental architecture maintains a context queue, which contains contexts from time $t-K$ to t, where K is a constant. At time t, a state $s(t)$ is determined by the current sensory input and the information in the last $K-1$ contexts. In other words, $s(t) = f'(l(t), l(t-1), \ldots, l(t-K+1))$, where f' is a function to derive $s(t)$ from contexts. We can also define $s(t) =$

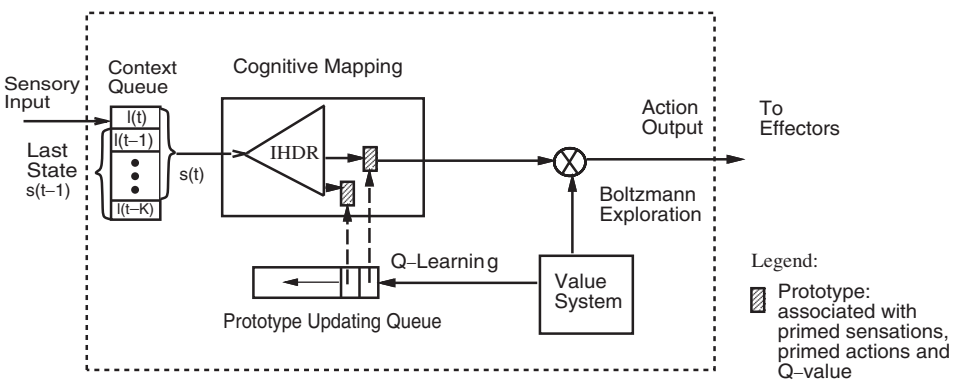


Fig. 1. The system architecture of developmental learning.

$f(l(t), l(t-1), l(t-2), \ldots, l(t-K)) = f(l(t), s(t-1))$. In other words, given the current context $l(t)$ and the last state $s(t-1)$, the agent can transfer to a new state. Mathematically, this is called observation-driven state transition: $f: L \times S \mapsto S$, where L is the context space, S is the state space. Section 2.1 describes how we define a state in this paper. Details of observation driven state transition can be found in Weng's paper.[37]

Cognitive mapping: it is the cognitive mapping module that maps the current state to the corresponding effector control signal. The cognitive mapping is realized by Incremental Hierarchical Discriminant Regression (IHDR).[39] A more detailed explanation is beyond our scope. Basically, given a state, IHDR finds the best matched s' associated with a list of primed contexts $[c' = (x', a', q)]$, which include: primed sensations $X' = (x'_1, x'_2, \ldots, x'_n)$, primed actions $A' = (a'_1, a'_2, \ldots, a'_n)$ and corresponding Q-values $Q = (q_1, q_2, \ldots, q_n)$, where n is the number of different actions. In Q-learning,[36] Q-value is defined as the expected discounted sum of future payoffs obtained by taking one action from a state. Details of Q-learning can be found in Sec. 3.3. In summary, the function of IHDR is $g: S \mapsto X' \times A' \times Q$. We should notice the "primed context" is different from "context" defined earlier.

Action selection: primed actions are the possible actions in each state. The probability of taking each primed action is based on its Q-value. The primed sensation predicts what the actual sensation will be if the corresponding primed action is taken. The value system works as an action selection function $v: 2^{A'} \mapsto A$ ($2^{A'}$ denotes all the possible subsets of A'), which chooses an action from a list of primed actions.

Novelty is measured by the difference between primed sensation and actual sensation. In the value system, novelty is integrated with reinforcement learning so that humans can issue rewards to modulate a developmental robot's behavior. In order to let the robot explore more states, Boltzmann Softmax exploration[33] is implemented. To reach the requirement of real-time and online updating in developmental learning, we add a prototype updating queue module to the architecture, which keeps the most recently visited states (indicated by dash lines). For example, if the length of the queue is 5, $s(t-5)$ to $s(t-1)$ will be saved in this queue. Only states in that queue are updated at each time instant so that updating can be done in real-time. In the following sections, we describe each component of the architecture in detail.

2.1. *Cognitive mapping and incremental hierarchical discriminant regression*

A detailed architecture of cognitive mapping is shown in Fig. 2. Initially, the sensory input updates the context queue, which includes multiple contexts $l(t)$. The length of the queue is $K+1$. In our experiments, $l(t) = \{x(t), p(t), a(t)\}$, that is, a context consists of current sensory input $x(t)$, neck position $p(t)$, and action

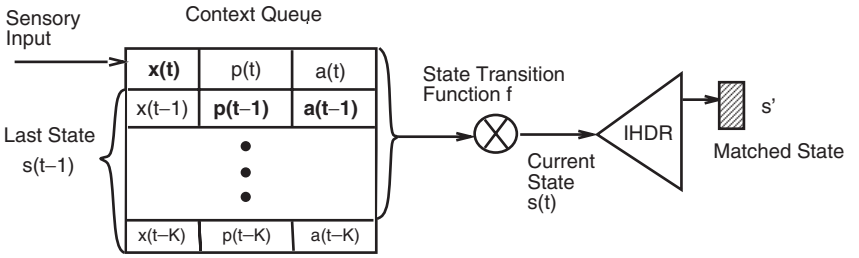


Fig. 2. Detailed architecture of cognitive mapping.

$a(t)$, where t is the time step. Note that this is a general architecture. We can choose different lengths of the context queue. For example, if this architecture is applied to speech recognition, the queue could be 20, which cover an utterance of about 0.4 s. In the experiments reported here, the length is two ($K = 1$). A state $s(t)$ in these experiments consists of two parts: visual image x and neck position p. According to the observation-driven state transition function, $s(t) = f(l(t), s(t-1))$, the last state provides information of the last neck position $p(t-1)$ and the last action $a(t-1)$. Based on these two items, we can calculate current neck position $p(t)$, which is combined with current visual input $x(t)$ to generate current state $s(t)$. The three factors: $x(t)$, $p(t-1)$, and $a(t-1)$ (in bold font in Fig. 2) determine $s(t)$.

What IHDR does is to find the best match s' of $s(t)$. IHDR automatically derives discriminating feature subspaces in a coarse-to-fine manner from a high-dimensional input space by generating a tree architecture of memory organization. At the root of the tree, a linear subspace is spanned by automatically derived discriminating features. The features are discriminative in the sense that input components that are irrelevant to the mapping's output are disregarded to achieve better discrimination and generalization (e.g. a facial mole distinguishes two like sisters). A probability-based nonlinear partition in the subspace divides the entire input space into a number of regions, each corresponding to a child node. Each child node receives input samples that belong to its region (e.g. human faces, but not exact) and it derives discriminating features subspaces (e.g. man and woman faces, but not exact) like its parent, but uses its assigned samples, and further partitions the subspace. Such a coarse-to-fine partition in the input space is carried out recursively until the node has received only a few samples (called states) and the node is then a leaf node (without children).

The relation of each matched state s' and its primed contexts is shown in Fig. 3. Besides primed contexts, a state consists of four kinds of information: age, learning rate, temperature, and standard deviation of the primed sensation. The age of a state is used to determine the learning rate and temperature of Boltzmann Softmax

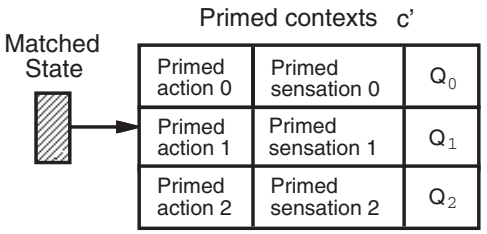


Fig. 3. State and its primed contexts. Each state is associated with a list of primed contexts. Each primed context consists of a primed action, a primed sensation and a Q-value.

exploration when the state is visited. The standard deviation is used to calculate novelty. More details can be found in the next section.

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. The value system of a robot at its "birth" time is called the innate value system. It further develops continuously throughout its "life" experience. The value system reported here integrates novelty and reinforcement learning.

3.1. *Novelty*

Habituation plays a very important role in non-associative learning.[21] Habituation is a short-term behavior of novelty.[41] Suppose the ith primed action is chosen at one state, we can define novelty as the normalized distance between the ith primed sensation $x'_i = (x'_1, x'_2, \ldots, x'_m)$ at time t and the actual sensation $x(t+1)$ at the next time:

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

where D 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 $(x'_j - x_j)^2$, as shown in Eq. (2):

$$\sigma_j^2(t) = \frac{t-1-\eta}{t}\sigma_j^2(t-1) + \frac{1+\eta}{t}(x'_j - x_j)^2, \tag{2}$$

where η is the amnesic parameter to give more weight to the new samples. With an appropriate η, $\sigma(t)$ would represent the short-term variation of the sensation. When a state is generated for the first time, the initial value of σ_i^2 is copied from its nearest neighbor in the same leaf node. The amnesic parameter is formulated by

Eq. (3):

$$\eta(t) = \begin{cases} 0 & \text{if } t \leq n_1, \\ c(t-n_1)/(n_2-n_1) & \text{if } n_1 < t \leq n_2, \\ c + (t-n_2)/m & \text{otherwise,} \end{cases} \tag{3}$$

where n_1 and n_2 are two switch points, c and m are two constant numbers which determine the shape of η. After the above calculations, $x'_j(t)$ would be set as $x_j(t+1)$, which is the new primed sensation.

How do we determine the value of the amnesic function $\eta(t)$? We design three intervals separated by two transition points, $t = n_1$ and $t = n_2$. For example, $n_1 = 20$, $n_2 = 200$. In the first interval, we would like to compute the straight incremental average for the maximal statistical efficiency (smallest error variance given the samples). Thus, we make $\eta(t) = 0$ when t changes from 0 to n_1. In the second interval, the number of samples is sufficient and we can afford to compute the amnesic average in order to gradually forget the old data. To compute the amnesic average when t changes from $t = n_1$ to $t = n_2$, we make $\eta(t)$ gradually grow until it reaches a constant c (e.g. $c = 1$, doubling the learning rate from the straight average or $c = 2$ to triple the rate). This amnesic weight for the new data $(\eta(t) + 1)/t$ will approach zero when t goes to infinity. This means that when t is extremely large, the new data would hardly be used and thus the system will hardly adapt. That is why we need the third interval for $t > n_2$, in which we would like to switch $\eta(t)$ to be a function of t. We would like to make $\eta(t)$ grow at an asymptotic rate of $1/m$ for a constant m, as t goes to infinity. Also we need to make $\eta(t)$ continuous at the point $t = n_2$. Note that, as t goes to infinity, the weight for the new data $x(t)$ is approximately the same as that of the non-amnesic average with m data points. Such a growing $\eta(t)$ enables the amnesic average to track the non-stationary random input process, whose mean changes slowly over time.

3.2. *Integration of novelty and rewards*

It is necessary to note that the novelty $n(t)$ 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 trainers and the environment can shape the robot's behaviors through punishments and rewards.

We introduce a concept of (innately) biased sensors to model sensors for which the value system has innate preference patterns at the birth time. Otherwise, a sensor is not (innately) biased. For example, a biased sensor value $r_g(t) = 1$ if the human teacher presses its "good" button (positively biased sensor) at time t and $r_b(t) = -1$ if the human teacher presses its "bad" button (negatively biased sensor) at time t. Furthermore, studies in animal learning show that different reinforcers have different effects. Punishment typically produces a change in behavior much more rapidly than other forms of reinforcers.[8]

We integrate novelty with immediate punishments and rewards so that the robot can take different factors into account. The combined reward is defined as a weighted

sum of physical reinforcers and the novelty:

$$r(t) = w_b r_b(t) + w_g r_g(t) + w_n n(t), \tag{4}$$

where $w_b > w_g > w_n$ are three normalized weights of punishment, reward and novelty, respectively, satisfying $w_b + w_g + w_n = 1$.

3.3. *Q-learning algorithm and Boltzmann softmax exploration*

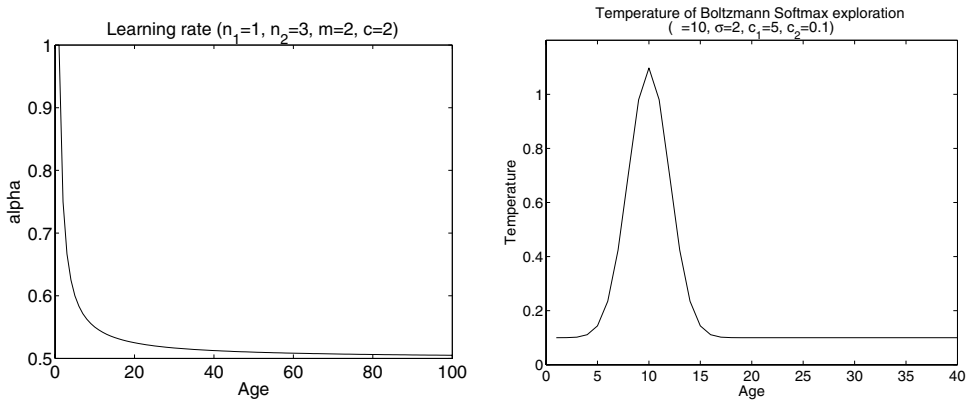
There are two major problems. First, the reward r is not always consistent. Humans may make mistakes in giving rewards. Thus, the relation between an action and the actual reward is not always certain. The second is the delayed reward problem. The reward due to an action is typically delayed since the effect of an action is not known until sometime after the action is complete. These two problems are dealt with by the Q-learning algorithm.[36] Q-learning is one of the most popular reinforcement learning algorithms. The basic idea is as follows. Each state s maintains a Q-value $[Q(s, c')]$ for every possible primed context c'. The action with the largest value will be selected as output and then a reward $r(t+1)$ will be received. We implemented a modified Q-learning algorithm as follows:

$$Q(s(t), c'(t)) \leftarrow (1-\alpha)Q(s(t), c'(t)) + \alpha(r(t+1) + \gamma Q(s(t+1), c'(t+1)), \tag{5}$$

where α and γ are two positive numbers between 0 and 1. $\alpha = (1+\eta)/t$ is a time varying learning rate based on amnesic average parameter (η). The parameter γ is for value discount in time. With this algorithm, Q-values are updated according to the immediate reward $r(t+1)$ and the next Q-value; thus, a delayed reward can be back-propagated in time during learning.

A major difference between this paper and traditional Q-learning algorithm is that we use changing learning rates based on amnesic average for each state instead of a global constant learning rate for the robot. The idea is derived from human development. At different maturity stages, the learning rules of human are different. A single value is not enough to model all the situations. For example, when we meet an unknown person for the first time, we would remember him right away (a high learning rate). Later, when we meet him dressed differently, we would gradually update his image in our brains with lower learning rates. The formulation of α guarantees that it has a large value at the beginning and converges to a constant smaller value through the robot's experience. Figure 4(a) shows an example of learning rate based on amnesic average. It is worth noting that in this figure, "Age" is the same as t.

We applied the Boltzmann Softmax exploration[33] to the Q-learning algorithm so that more states could be visited. Furthermore, this implementation of Q-learning also shows the difference between the symbolic world and the real world. In traditional Q-learning models, we choose an action with the highest Q-value. However, in the real world, this strategy cannot guarantee to reach the best state. One action can lead to one of many possible states. In AI, this is called the qualification and



(a) (b)

Fig. 4. (a) Learning rate based on amnesic average (the parameters are shown in the title); (b) Temperature of Boltzmann Softmax exploration based on Guassian density model. Age is the same as time (t).

ramification problem.[24] We should not use the "highest" Q-value as the traditional Q-learning does. Instead, we have to take into account the above problem and implement the modified Q-learning algorithm. At each state (s), the robot has a list of actions $A(s) = (a'_1, a'_2, \ldots, a'_n)$ to choose from. The probability of action a being chosen at s is:

$$p(s, a') = \frac{e^{\frac{Q(s,a')}{\tau}}}{\Sigma_{a' \in A(s)} e^{\frac{Q(s,a')}{\tau}}}, \tag{6}$$

where τ is a positive parameter called temperature. With a high temperature, all actions in $A(s)$ have almost the same probability of being chosen. When $\tau \to 0$, the Boltzmann Softmax exploration more likely chooses action a' that has a high Q-value.

The question is how to determine τ. In our early work,[10] we used a constant value. However, it causes problems when the value system is applied to the real world. As we know, when we sense a novel stimulus at the first time, we would pay attention to it for a while. In this case, a small τ is preferred because the Q-value of action "stare" would be high and the robot should choose this action. If τ is too large, the probability of each action is almost equal, which is not the case under attention. After staring at the novel stimulus for a while, the robot would feel tired and pay attention to other stimuli. Now a larger τ is preferred. After a period of exploration τ should drop again, which means the state is fully explored, the robot can take the action with the highest Q-value. If we choose a large constant τ, then the robot would explore even though it visits a state for the first time. If we choose a small τ, the robot would face the local minimal problem and cannot explore enough states. Fortunately, a Guassian density model [Eq. (7)] for local temperature solves

this problem:

$$\tau(t) = \frac{c_1}{(2\pi)^{1/2}\sigma}\exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right] + c_2, \tag{7}$$

where c_1 is a constant to control the maximal value of temperature, c_2 controls the minimal value, and t is the age of the state. The plot of the model can be found in Fig. 4(b).

As discussed in the book *Reinforcement Learning — An Introduction*,[33] the conditions required to assure convergence with probability one are: $\sum_{t=1}^{\infty} \alpha(t) = \infty$ and $\sum_{t=1}^{\infty} \alpha^2(t) < \infty$. "The first condition guarantees that the learning rates are large enough to eventually overcome any initial conditions or random fluctuations. The second condition guarantees that eventually the learning rates become small enough to assure convergence." We use $\alpha(t)$ to denote the learning rate at time t. In our case, the learning rate $[\alpha(t)]$ changes from 0.5 to 1 [see Fig. 4(a)] because of the Guassian density model [Eq. (7)]. The second condition is not met, indicating that the estimates cannot completely converge but continue to vary in response to the most recently received rewards. But this property is actually desirable in a nonstationary environment. For a developmental robot that has to explore in unknown environments, the problems it faces are effectively nonstationary. Thus, a developmental robot will not be limited to a given fixed task-specific goal, but to learn autonomously in many different settings, through its experience in the world.

3.4. *Prototype updating queue*

In the batch learning mode of a reinforcement learning algorithm, back-up 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 is why we designed the prototype updating queue in Fig. 1, which stores the addresses of formerly visited states. In this queue, Q-value is back-propagated, so is the primed sensation. This back-up is performed iteratively from the tail of the queue back to the head of the queue. After the entire queue is updated, the current state's address is pushed into the queue and the oldest state at the head is pushed out of the queue. Because we can limit the length of the queue, real-time updating becomes possible. Watkins's $Q(\lambda)$ (eligibility trace)[36] does not look ahead all the way to the end of the episode in its backup but look ahead as far as the next exploratory action. However, in the value system reported here, we implemented Boltzmann Softmax exploration, which means that if we use the traditional eligibility trace, all of the traces would be set to zero. Then the eligibility trace is useless. In contrast, PUQ does not discriminate between exploratory and greedy actions but looks ahead all the way through the queue.

3.5. *Algorithm of the inherent value system*

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

(i) Grab the new sensory input $x(t)$ to update context $l(t)$; combine $l(t)$ with last state $s(t-1)$ to generate current state $s(t)$.
(ii) Query the IHDR tree and get a matched state s' and related primed contexts.
(iii) If $s(t)$ is significantly different from s', it is considered as a new state and the IHDR tree is updated by saving $s(t)$. Otherwise, use $s(t)$ to update s' through incremental averaging.
(iv) Update the age of the state, calculate the temperature of the state with Eq. (7).
(v) Use the Boltzmann Softmax Exploration in Eq. (6) to choose an action based on the Q-value of every primed action. Execute the action.
(vi) Calculate novelty using Eq. (1) and integrate with immediate reward $r(t+1)$ using Eq. (4).
(vii) Update the learning rate based on amnesic average.
(viii) Update the Q-value of states in PUQ. Go to Step (i).

4. Simulation

In order to test the value system, a simulation environment is developed. The simulator GUI is shown in Fig. 5. The big window shows the viewing environment, while

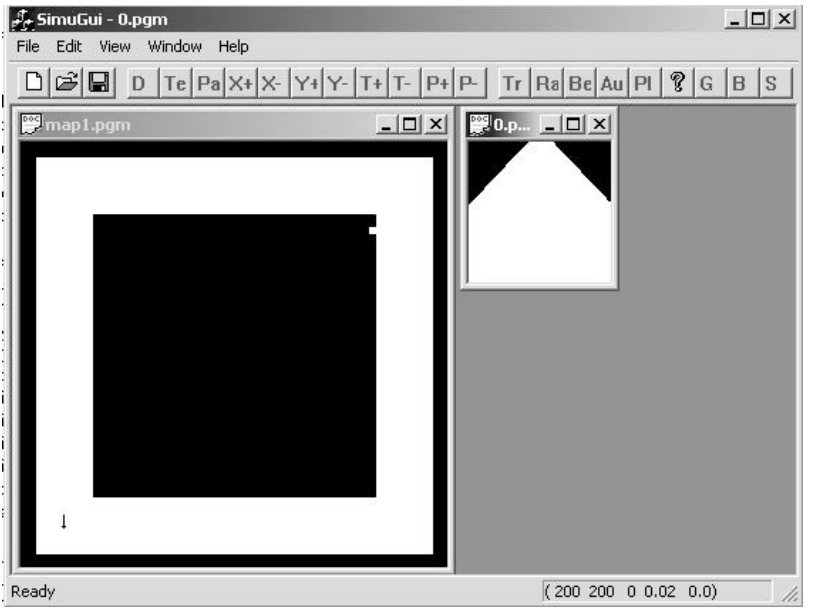


Fig. 5. The GUI simulator. The arrow indicates the position and the viewing angle of the robot.

the small window shows the image 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: action 0 (stay at the current viewing angle), action 1 (turn neck left 30°) and action 2 (turn neck right 30°). The representation of a state consists of visual images and absolute viewing angle. We assume the robot can only turn its neck at sampled positions. The angle between its head and the body could be $-90°, -60°, -30°, 0°, 30°, 60°$, and 90°. Each position corresponds to an absolute view angle. We assume that the robot cannot look backward and the number of absolute viewing angle is 7. If the robot faces the front, the absolute view angle is 0. The dimension of input image is 100×100. The parameters are defined as follows: $\gamma = 0.5$ in Eq. (5); $c = 2$, $n_1 = 1$, $n_2 = 3$, $m = 2$ in Eq. (3). The value of c_1 in Eq. (7) is 5; $c_2 = 0.1$; $\mu = 20$ and $\sigma = 2$.

In the first experiment (Sec. 4.1), we let the agent explore by itself (about 300 steps). At the beginning, novelty was high and the agent kept exploring. Gradually, it experienced all the possible visual inputs and did not have a preference for any action. An IHDR tree was saved at that point. Then the trainer issued rewards to action 1, which became dominant (Sec. 4.2). In the third experiment (Sec. 4.3), a moving object was added to the scene, the agent chose action 0 and stared at this object because of high novelty. And finally (Sec. 4.4), the trainer issued punishment to action 0 and rewards to action 2. Gradually, the agent chose action 2 most of time. Details of each experiment are as follows.

4.1. *Habituation effect*

In the first experiment, we allowed the robot to explore on its own by looking around. The total number of state is equal to the number of view angle (7) because the input image is always the same at each view angle. If one view is really boring, the robot turns its head away. The initial Q-value of each action is 0. Figure 6 shows how the Q-value of each action changes based on novelty in one state (the absolute view angle of the state is 0 and the input image of the state is shown in the small window of Fig. 5).

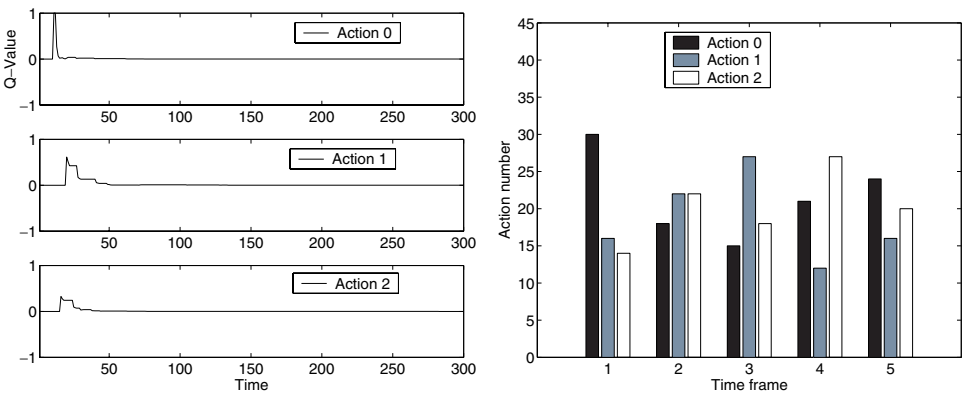
Initially, the primed sensation is set as a long vector in which every element is zero. After taking an action, the current sensation is very different from the initial sensation. That is, the novelty value is high. We can see from Fig. 6 that the Q-values of each action increase during the first several steps. However, after the primed sensation is updated, it would be the same as the actual sensation if the robot takes the action again. Then, the novelty becomes zero and the Q-value decreases. After a period of training (100 steps), the robot can predict the actual sensation of the next step very well. So the Q-value of each action converges to the same value (0). This means each action has the same probability of being chosen. The right-hand side of Fig. 6 shows the number of each action in different time frame. We divided 300 steps into 5 time frames. Each time frame consists of 60 steps. The

numbers of different actions are not equal even after exploration because we used Boltzmann Exploration to randomly generate actions. It is unlikely to get equal numbers for each action. However, no action is dominant. After 100 steps, Q-values of different actions are nearly equal, so the numbers of different actions are close. The experiment shows that because of the habituation effect, the robot was not interested in any action after exploration and chose an action randomly. Because a state is not visited at every time instant, there are flat periods in the Q-value plot, which means that the robot is not at the state and the Q-value does not change.

4.2. *Integration of novelty and positive reward*

After the above experiment, we began to issue rewards. For example, when the robot took action 1, a human teacher gave it a positive reward (1). For action 2, negative rewards (−1) were issued. Then the actual reward the robot receives is an integration of novelty and immediate rewards. The Q-value of action 0 converges to 0. That of action 2 becomes negative after a punishment was issued at step 348. The Q-value of action 1 is always positive because we kept issuing positive rewards. As we can see on the left-hand side of Fig. 7, after training, the Q-value of action 1 is much larger than that of other actions. The number of each action is shown in Fig. 7 (right). Compared to Fig. 6 (right), action 1 was dominant.

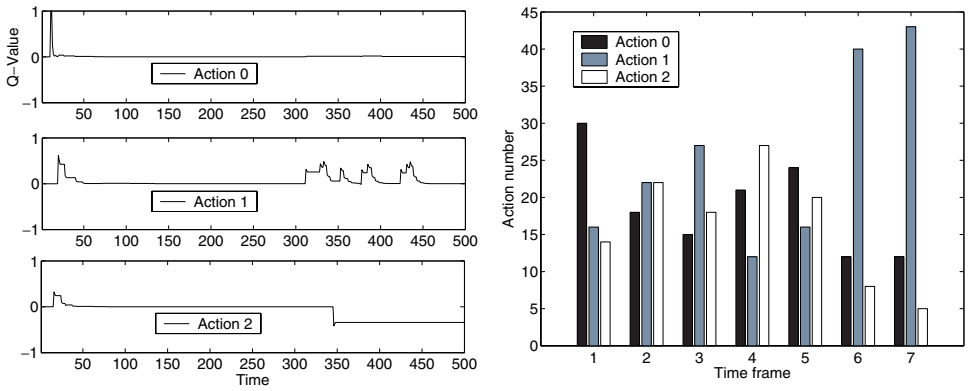
If one action is stimulated by novelty and another action is stimulated by a positive reward, then the latter action will predominate because in the value system the priority of reward is higher than that of novelty [according to Eq. (4), reward has a larger weight than novelty].



Q-value versus time

The number of each action

Fig. 6. Habituation effect. On the left-hand side, the first, second and third plots correspond to the Q-value of action 0, action 1, and action 2, respectively. The x-axis is time step; the y-axis is the Q-value. On the right-hand side, we compare the numbers of different actions in each time frame. Each time frame consists of 60 steps.



Q-value versus time

The number of each action

Fig. 7. Integration of novelty and immediate reward. On the left-hand side, the first, second and third plots correspond to the Q-value of action 0, action 1, and action 2, respectively. The x-axis is the time step; the y-axis is the Q-value. On the right-hand side, we compare the numbers of different actions in each time frame. Each time frame consists of 60 steps.

Fig. 8. Simulation of a moving object.

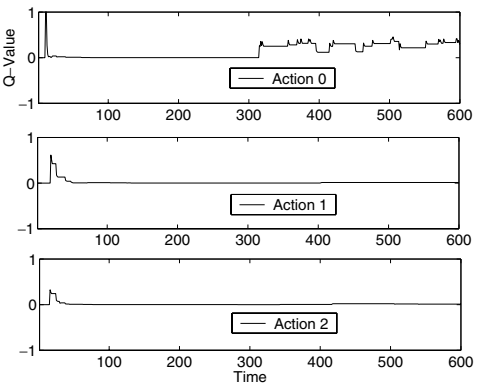
4.3. *Increase novelty with a moving object*

In order to show novelty preference, a moving toy was added to the simulation environment after the first experiment. The test images are shown in Fig. 8.

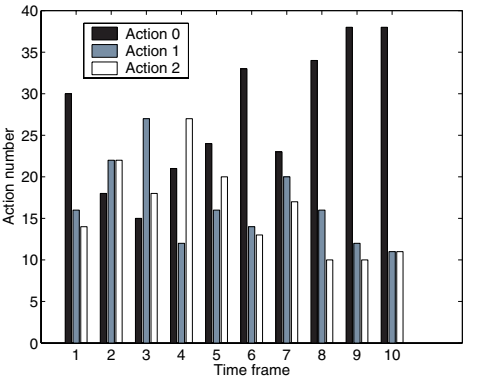
There are five different images of a toy. Every time when the robot's absolute viewing angle is 0, one of these images is generated randomly as the visual input. Thus, at this position the primed sensation of action 0 could be different from the actual sensation. In the last two experiments, the number of states is 7. Now the total number of states is 12. As shown on the left-hand side of Fig. 9, the Q-value of action 0 is positive because of high novelty value. In contrast, the Q-values of actions 1 and 2 are close to zero. After training, the robot figured out that staying in the current state is the most interesting. So action 0 was chosen the most often.

4.4. *Suppress novelty with punishment*

After the third experiment, we issued positive rewards to action 2, 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 increased. As shown in Fig. 10, after training, the robot chose

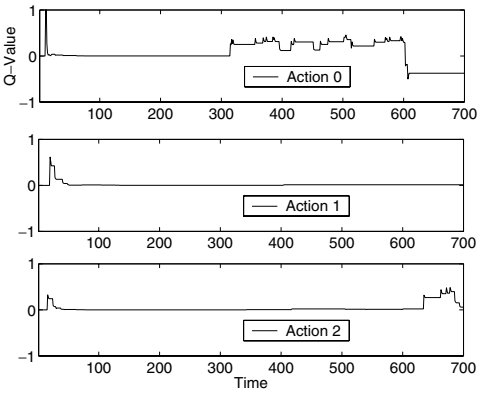


Q-value versus time

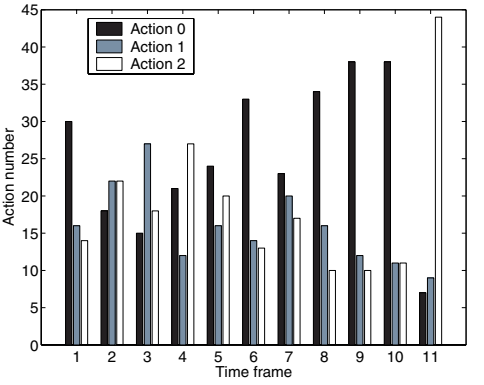


The number of each action

Fig. 9. Increase novelty with a moving object. On the left-hand side, the first, second and third plots correspond to the Q-value of action 0, action 1, and action 2, respectively. The x-axis is the time step; the y-axis is the Q-value. On the right-hand side, we compare the numbers of different actions in each time frame. Each time frame consists of 60 steps.



Q-value versus time



The number of each action

Fig. 10. Suppress novelty with immediate rewards. On the left-hand side, the first, second and third plots correspond to the Q-value of action 0, action 1, and action 2, respectively. The x-axis is the time step; the y-axis is the Q-value. On the right-hand side, we compare the numbers of different actions in each time frame. Each time frame consists of 60 steps.

action 2 most of the time. However, actions 0 and 1 were still chosen a few times because of Boltzmann Softmax exploration.

5. Experiments with SAIL Robot

The value system was also tested on our SAIL robot (short for Self-organizing Autonomous Incremental Learner) through vision-guided neck action selection. SAIL, shown in Fig. 11, is a human-size robot at Michigan State University. It

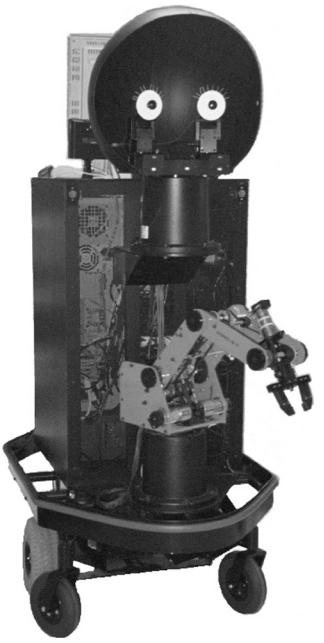


Fig. 11. SAIL robot at Michigan State University.

has two "eyes," which are controlled by fast pan-tilt heads. In the real-time test, at each step SAIL (placed in a lab) has three action choices: action 0 (stay), action 1 (turn its neck left), and action 2 (turn its neck right). In total, there are seven absolute positions of its neck. The center is position 0, and from left to right are positions -3 to 3. Even though the degree of freedom is only one, the difficulty lies in the uncontrolled environment. Because there is a lot of noise in the real-time test (people come in and out), we restricted the number of states (about 60) by applying a Gaussian mask to image input after subtracting the image mean. The dimension of the input image is $30 \times 40 \times 3 \times 2$, where "3" arises from RGB colors and "2" for two eyes. A subset of input images are shown in the first row of Fig. 13; different toys are used as novel stimuli. The state representation consists of visual image and the absolute position of the robot's neck. These two components are normalized so that each has similar weight in the representation. Biased touch sensors are used to issue punishment (value is set to be -1) and reward (value is set to be 1). Since we applied a Gaussian mask to each input image, background noise can be removed so that the robot will not be distracted by minor visual changes. For other parameters, we used the same values as we did in the simulation.

5.1. *Novelty and multiple reinforcers for different actions*

In order to show the effect of novelty, we allowed the robot to explore by itself for about 5 min (200 steps), then kept moving toys at neck position -1. At each

position, there could be multiple states because the input images at certain neck positions could change. Figure 12 shows the information of one state at position -1. The image part of the state is the fourth image of the first row shown in Fig. 13, which is the background of the experiment. The first three plots are the Q-value of each action, the fourth plot is the reward of each action, the fifth plot is the novelty value, and the last one is the learning rate of the state. The learning rate is determined by how many times the state has been visited rather than the absolute time. That is why it is plotted at discrete time intervals. Action $0, 1, 2$ are specified by ".," "*," "+," respectively. After exploration (200 steps later), we

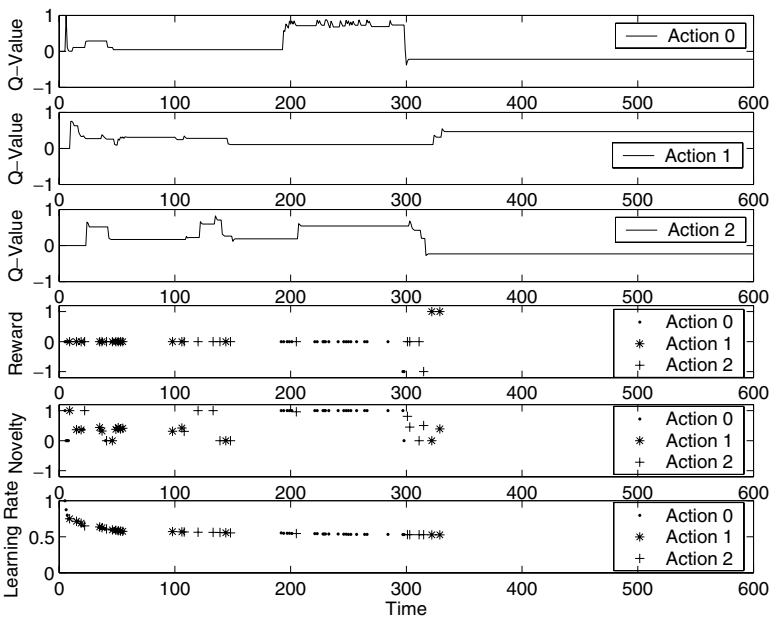


Fig. 12. The Q-value, reward, novelty and learning rate of each action of one state at position -1 when multiple reinforcers are issued.

Fig. 13. Part of a sample sensation sequence. The first row is the image sequence captured by the robot. The second row is the actual visual sensation input after Gaussian windowing. The corresponding primed visual sensation is shown in the third row.

moved toys in front of the robot, which increases the novelty and Q-value of action 0. After training, the robot preferred toys and kept looking at them from step 230 to step 270.

A subset of the image sequence is shown in Fig. 13. The first row is the image sequence captured by the robot. The second row is actual visual sensation sequence after applying the Gaussian mask. The corresponding primed visual sensation is shown in the third row. If the actual sensations in the second row and the corresponding primed sensation in the third row are very different, the novelty would be high. After step 300, the trainers began to issued different reinforcers to different actions. Punishments were issued to action 0 at step 297 and step 298 (the fourth plot) and to action 2 at step 315. Rewards were issued to action 1 at step 322 and step 329. The Q-values of actions 0 and 2 became negative while that of action 1 became positive, which means that the visual attention ability of the robot is developed through the interactions with the environment. Even though the novelty of action 0 is higher, the robot preferred action 1 because of its experience. The learning rate in the fifth row shows that at the beginning the robot immediately memorized the new stimuli and then gradually updated the stimuli.

5.2. *Boltzmann softmax exploration*

As we mentioned in Sec. 3, Boltzmann Softmax exploration is applied so that the robot can experience more states. In Fig. 14, only information from step 1 to step 60 of the above state described in Sec. 5.1 is shown. The first plot is the probability of each action based on its Q-value. The total probability is 1. The probabilities of action 0, 1, 2 are plotted at the top, middle and bottom, respectively. The star denotes the random value generated by a uniform distribution. If the random value is in one range (say, the middle range), then corresponding action (say, "action 1") would be taken. Because the robot is not always in the state, the plot is sparse. The second plot shows the temperature based on the Gaussian density model [Eq. (7)].

At the beginning, τ is small and the novelty of the state is high (the initial Q-values of other actions are zero), so the probability of action 0 is the largest (almost 100%). The robot stared at the stimulus for while. Then, the temperature increased. The probabilities of each action became similar and the robot began to choose other actions and explored more states. After about ten visits, the temperature dropped to a small value (0.1) again; the action with a larger Q-value would have a higher probability of being taken. As we mentioned in Sec. 3.3, on the one hand, if τ is defined as a large constant, the robot would keeping exploring. On the other hand, if τ is defined as a small constant, the robot would get stuck in the local minimal problem. Fortunately, with τ defined in Eq. (7), we solved the problem.

The histogram of the age of all states is shown in Fig. 15. Most states were only visited fewer than 20 times, which is because we kept moving different toys in front of the robot and could not guarantee to put the toy at the same direction and position.

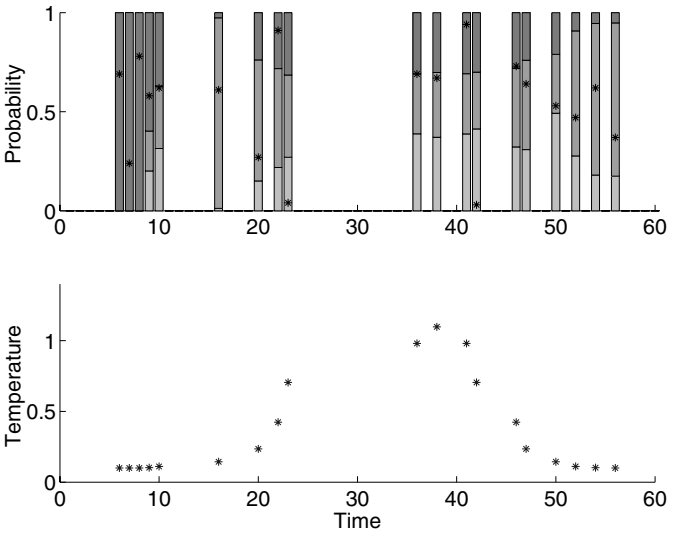


Fig. 14. Boltzmann Softmax Exploration. The top plot shows the probability of each action based on its Q-value. The probabilities of action $0, 1, 2$ are plotted at the top, middle and bottom areas, respectively. The probability is represented by the length of a bar. The total probability is 1. The star denotes the random value (between 0 and 1) received at that time. The robot chooses an action when the star fits into the corresponding range. The bottom plot shows the temperature.

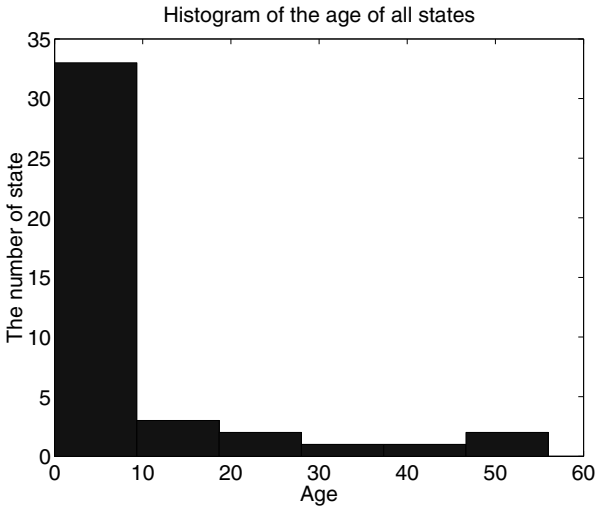


Fig. 15. Histogram of the age of all states.

6. Discussion

There are several major limitations in the scope of the proposed architecture. First, although introducing sensory novelty in the motivational system requires a very challenging high-dimensional cortical prediction capability (simulated by IHDR), further work is needed to show how this three-component inherent motivational system facilitates developmental learning of more complex tasks and multiple tasks. For example, in this paper the action space is only one-dimensional, which is a restricted setting. In principle, we can add more degrees of freedom (e.g. the tilt degree) and use the same architecture to train the robot because IHDR can learn in a high-dimensional space. However, after we increase the degrees of freedom, the state space will increase. In this case, it would take more time to train the robot or let it explore. The major disadvantage of reinforcement learning based architecture is that in general many trials are required to learn an optimal action. How to speed up learning for developmental robots is still an open question.

Second, the current presented experiments involve only low-level sensorimotor behaviors. In order to develop higher intelligence, the inherent motivational system must facilitate a more advanced type of learning — communicative learning. With the communicative learning, the human teacher can directly state: (i) a desired action in the current context; (ii) whether the current action is good; (iii) the rules to follow in order to reach desired actions (as in animal training and classroom teaching); (iv) the criteria to judge right or wrong; success or failure (teaching the value system) (see Ref. 3 for more discussion).

Third, the measurement of development is very important. Although the proposed value system has been shown to interact with three components (punishment, reward and novelty) in uncontrolled environments, there are many open questions. For example, how does the value system develop to a larger scale other than what is possible with the Q-learning mechanism? A value system that motivates mental development should grow in the developmental process. That is why we implement the proposed value system using an IHDR tree. The complexity of such a development is reflected in the tree. Initially, it only has a few states. After exploring and interacting with human teachers, the tree grows and the performance of the motivational system improves. However, the developmental robotics field still lacks studies on how to evaluate development. A field called psychometrics (e.g. a practical scale[5] used by clinical psychologists) has developed systematic scales for measuring human cognitive capabilities. It could be a measurement of value system complexity.

Fourth, the field of developmental robotics has reached a point to where it shows great promise of development for application domains with multimodal inputs. Currently, in some of our projects, the developmental robots develop skills for multiple sensory inputs (vision, speech, touch) (see some related projects in Ref. 37). For instance, the value system can guide a robot to navigate in different environments with visual input. At the same time, the teacher can use "bad" and "good" touch sensors (reinforcement learning) and verbal commanding (communicative learning) to shape the robot's navigating behaviors.

7. Conclusions

While a lot can be done along the direction of developmental value systems, the proposed value system is the first developmental model for a general-purpose inherent value system that integrates punishment, reward and novelty as far as we know. The context novelty is derived from the high-dimensional primed (predicted) sensation, which is enabled by the IHDR engine. Thus, "value" is not just a scalar number as in many reinforcement learning studies. Instead, it characterizes the rich information of the environment. Such an integrated value system can guide a developmental robot all the time since novelty from primed sensation is densely defined at every sensory refresh cycle. Because no salient features are predefined, the value system is applicable, in principle, to many settings. The combination of punishment, reward and novelty is a starting point for the unification of different types of animal learning, such as classical conditioning, instrumental conditioning, habituation, and other value-driven learning such as attention selection. The proposed inherent value system was successfully tested on an agent in the simulation environment and a robot in the real world. This framework not only allows autonomous exploration of the environment by the robots but also enables the environment (including human teachers) to shape the sensorimotor behaviors by providing punishment and reward.

As discussed in the last section, a very powerful mechanism is to enable this inborn value system to develop complex skills for communicative learning. For example, with this inherent value system, we can verbally teach the robot what "good" and "bad" are. After the robot acquires these words and their associations with reward (or punishment), we can train the robot through communicative learning to develop higher intelligence. Another challenge is to measure the complexity of a value system. It seems that punishment, reward and novelty are necessary for a developmental agent including humans. Are there any other necessary mechanisms for the human inherent value system? How can we find a set of developmental mechanisms for the inherent value system that is both necessary and sufficient to develop intelligence at different levels? This is a totally new topic. We know that no two humans have exactly the same voice, but they can pronounce semantically equivalent words. Psychologists have extensively studied human categorical perception and developed psychometrics to measure cognitive capabilities. It is desirable to consider psychometrics in the search for the necessary and sufficient set of mechanisms for the inherent value system. There are still plenty of practical and theoretical questions awaiting further investigation. We hope this work opens up a wide range of opportunities for developmental learning.

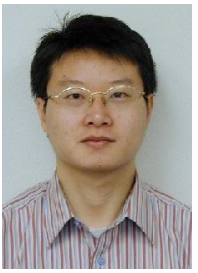
Acknowledgments

The authors would like to thank Wey S. Hwang for his major contribution to an earlier version of the IHDR program. The work is supported in part by the 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.

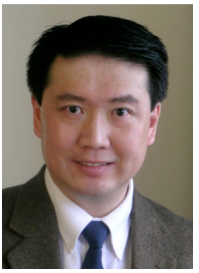
References

1. N. Almassy, G. M. Edelman and O. Sporns, Behavioral constraints in the development of neural properties: A cortical model embedded in a real-world device, *Cereb. Cortex* **8** (1998) 346–361.
2. M. Asada, K. MacDorman, H. Ishiguro and Y. Kuniyoshi, Cognitive developmental robotics as a new paradigm for the design of humanoid robots, *Robot. Autonom. Syst.* **37** (2001) 185–193.
3. M. Asada, S. Noda, S. Tawaratsumida and K. Hosoda, Purposive behavior acquisition on a real robot by vision-based reinforcement learning, *Mach. Learn.* **23** (1996) 279–303.
4. A. Barto, S. Singh and N. Chentanez, Intrinsically motivated learning of hierarchical collections of skills, in *Proc. 3rd Int. Conf. on Development and Learning*, La Jolla, California, 20–22 October 2004 (IEEE Press, 2004), pp. 112–119.
5. N. Bayley, *Bayley Scales of Infant Development* (Psychological Corp, San Antonio, TX, 1993).
6. V. Castellugcci and E. Kandel, An invertebrate system for the cellular study of habituation and sensitization, in *Habituation: Perspective from Child Development, Animal Behavior and Neurophysiology*, eds. T. Tighe and R. Leaton (Lawrence Erlbaum Associates, Hillsdale, NJ, 1976), pp. 1–48.
7. L. Cohen, Habitutation of infant visual attention, in *Habituation: Perspective from Child Development, Animal Behavior and Neurophysiology*, eds. T. Tighe and R. Leaton (Lawrence Erlbaum Associates, Hillsdale, NJ, 1976), pp. 207–238.
8. M. Domjan, *The Principles of Learning and Behavior* (Brooks/Cole Publishing Company, Belmont, CA, 1998).
9. J. H. Flavell, P. H. Miller and S. A. Miller, *Cognitive Development* (Prentice-Hall, Englewood Cliffs, NJ, 1993).
10. X. Huang and J. Weng, Novelty and reinforcement learning in the value system of developmental robots, in *2nd Int. Workshop on Epigenetic Robotics*, Edinburgh, Scotland (10–11 August 2002), pp. 47–55.
11. L. Itti, C. Koch and E. Niebur, A model of saliency-based visual attention for rapid scene analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(11) (1998) 1254–1259.
12. S. Kakade and P. Dayan, Dopamine bonuses, in *Advances in Neural Information Processing Systems*, Vol. 13, Denver, CO (MIT Press, 2000), pp. 131–137.
13. S. Kakade and P. Dayan, Dopamine: Generalization and bonuses, *Neural Network* **15** (2002) 549–559.
14. P. W. Kaplan, J. S. Werner and J. W. Rudy, Habitutation, sensitization and infant visual attention, in *Advances in Infancy Research*, eds. C. Rovee-Collier and L. Lipsit (ABLEX Publishing Corporation, Norwood, NJ, 1990), pp. 61–110.
15. C. Koch and S. Ullman, Shifts in selective visual attention towards the underlying neural circuitry, *Human Neurobiol.* **4** (1985) 219–227.
16. M. Lungarella, G. Metta, R. Pfeifer and G. Sandini, Developmental robotics: A survey, *Connection Sci.* **15**(4) (2003) 151–190.
17. S. Marsland, U. Nehmzow and J. Shapiro, Novelty detection on a mobile robot using habituation, in *From Animals to Animats, 6th Int. Conf. Simulation of Adaptive Behaviour*, Paris, France (2000), pp. 188–198.
18. H. Ogmen, A developmental perspective to neural models of intelligence and learning, in *Optimality in Biological and Artificial Networks?* eds. D. Levine and R. Elsberry (Lawrence Erlbaum Associates, Hillsdale, NJ, 1997), pp. 363–395.
19. P. Y. Oudeyer and F. Kaplan, Intelligent adaptive curiosity: A source of self-development, in *Proc. 4th Int. Workshop on Epigenetic Robotics*, Genoa, Italy (25–27 August 2004), pp. 127–130.

20. P. Y. Oudeyer and F. Kaplan, The playground experiment: Task-independent development of a curious robot, in *Proc. AAAI Spring Symp. Workshop on Developmental Robotics*, Stanford, CA (2005), pp. 42–47.
21. H. Peeke and M. Herz, *Habituation* (Academic Press, 1973).
22. J. Piaget, *The Origins of Intelligence in Children* (International Universities Press, New York, 1952).
23. M. Rucci, G. Tononi and G. M. Edelman, Registration of neural maps through value-dependent learning: Modeling the alignment of auditory and visual maps in the barn owl's pitic tecturm, *J. Neurosci.* **17** (1997) 334–352.
24. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall, Englewood Cliffs, NJ, 1995).
25. B. Scassellati, Theory of mind for a humanoid robot, *Autonom. Robot.* **12** (2002) 13–24.
26. J. Schmidhuber, Curious model-building control systems, in *Proc. Int. Joint Conf. Neural Networks*, Singapore (IEEE Press, 1991), pp. 1458–1463.
27. W. Schultz, Multiple reward signals in the brain, *Nature Rev. Neurosci.* **1** (2000) 199–207.
28. O. Sporns, Modeling development and learning in autonomous devices, in *Workshop on Development and Learning*, E. Lansing, Michigan, USA, April 5–7 (2000), pp. 88–94.
29. O. Sporns, N. Almassy and G. Edelman, Plasticity in value system and its role in adaptive behavior, *Adaptive Behav.* **8**(2) (2000) 129–148.
30. J. Stanley, Computer simulation of a model of habituation, *Nature* **261** (1976) 146–148.
31. M. Sur, A. Angelucci and J. Sharm, Rewiring cortex: The role of patterned activity in development and plasticity of neocortical circuits, *J. Neurobiol.* **41** (1999) 33–43.
32. R. S. Sutton and A. G. Barto, Toward a modern theory of adaptive networks: Expectation and prediction, *Psychol. Rev.* **88**(2) (1981) 135–170.
33. R. S. Sutton and A. G. Barto, *Reinforcement Learning — An Introduction* (MIT Press, Cambridge, MA, 1998).
34. R. Thompson, The neurobiology of learning and memory, *Science* **233** (1986) 941–947.
35. S. Thrun, *Exploration in Active Learning* (MIT Press, Cambridge, MA, 1995).
36. C. J. Watkins, Qlearning, *Mach. Learn.* **8** (1992) 279–292.
37. J. Weng, Developmental robotics: Theory and experiments, *Int. J. Humanoid Robot.* **1** (2004) 199–236.
38. J. Weng, From neural networks to the brain: Autonomous mental development, *IEEE Comput. Intell. Mag.* **1**(3) (2006) 15–31.
39. J. Weng and W. Hwang, An incremental learning algorithm with automatically derived discriminating features, in *Proc. 4th Asian Conf. Computer Vision*, Taipei, Taiwan, 8−9 January 2000 (Elsevier Science, 2000), pp. 426–431.
40. J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur and E. Thelen, Autonomous mental development by robots and animals, *Science* **291** (2000) 599–600.
41. D. Zeaman, The ubiquity of novelty-familiarity effects, in *Habituation: Perspective from Child Development, Animal Behavior and Neurophysiology*, eds. T. Tighe and R. Leaton (Lawrence Erlbaum Associates, Hillsdale, NJ, 1976), pp. 297–320.



Xiao Huang received his Ph.D. degree in Computer Science from Michigan State University, Lansing, Michigan, USA, in 2005. He is currently a software engineer at Adapx (formerly Natural Interaction Systems, LLC). His research interests include human-computer interaction, computer vision, augmented reality, machine learning and intelligent robots.



Juyang Weng is a Professor at the Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan, USA. His research interests include biologically inspired neural systems, vision, audition, touch, human-machine multimodal interface, and intelligent robots. He is an Editor-in-Chief of International Journal of Humanoid Robotics and the Chairman of the Governing Board of the multidisciplinary International Conferences on Development and Learning (ICDL) (http://cogsci.ucsd.edu/∼triesch/icdl). He was the Chairman of the Autonomous Mental Development Technical Committee of the IEEE Computational Intelligence Society (2004-2005), an Associate Editor of IEEE Transactions on Pattern Recognition and Machine Intelligence, an Associate Editor of IEEE Transactions on Image Processing. He initiated and supervised the SAIL (Self-organizing Autonomous Incremental Learner) and Dav projects, in which he and his co-workers designed and custom built robots for research on robotic computational realization of autonomous mental development. More details are available online at http://www.cse.msu.edu/∼weng.