

Developmental Robots: A New Paradigm *

Juyang Weng

Yilu Zhang

Department of Computer Science and Engineering

Michigan State University

East Lansing, MI 48824

{weng, zhangyil}@cse.msu.edu

Abstract

It has been proved to be extremely challenging for humans to program a robot to such a sufficient degree that it acts properly in a typical unknown human environment. This is especially true for a humanoid robot due to the very large number of redundant degrees of freedom and a large number of sensors that are required for a humanoid to work safely and effectively in the human environment. How can we address this fundamental problem? Motivated by human mental development from infancy to adulthood, we present a theory, an architecture, and some experimental results showing how to enable a robot to develop its mind automatically, through online, real time interactions with its environment. Humans mentally “raise” the robot through “robot sitting” and “robot schools” instead of task-specific robot programming.

1. Introduction

The conventional mode of developmental process for a robot is not automatic – a human designer is in the loop. A typical process goes like this: Given a robotic task, it is the human designer who analyzes and understands the task. Based on his understanding, he comes up with a representation, chooses a computational method, and writes a program that implements his method for the robot. The representation reflects very much the human designer’s understanding of the robot task. During this developmental process, some machine learning might be used, during which some parameters are adjusted according to the collected data. However, these parameters are defined by the human designer’s representation for the given task. The resulting program is for this task only, not for any other tasks. If the robotic task is complex, the capability of handling variation of environment is very much limited by the human designed task-specific representation. This manual development paradigm has met tremendous difficulties for

tasks that require complex cognitive and behavioral capabilities, such as many sensing and behavioral skills that a humanoid must have in order to execute human high-level commands, including autonomous navigation, object manipulation, object delivery, target finding, human-robot interaction through gesture in unknown environment. The high degree of freedom, the redundant manipulators, and the large number of effectors that a humanoid has, plus the multimodal sensing capabilities that are required to work with humans further increase the above difficulties. The complex and changing nature of human environment has made the issue of autonomous mental development of robots — the way human mind develops — more important than ever.

Many robotics researchers may believe that human brain has an innate representation for the tasks that humans generally do. However, recent studies of brain plasticity have shown that our brain is not as task-specific as commonly believed. There exist rich studies of brain plasticity in neuroscience, from varying extent of sensory input, redirecting input, transplanting cortex, to lesion studies, and sensitive periods. Redirecting input seems illuminating in explaining how much task-specific our brain really is. For example, Mriganka Sur and his coworkers rewired visual input to primate auditory cortex early in life. The target tissue in the auditory cortex, which is supposed to take auditory representation, was found to take on *visual* representation instead (Sur et al., 1999). Furthermore, they have successfully trained the animals to form visual tasks using the rewired auditory cortex (von Melchner et al., 2000). Why are the self-organization schemes that guide development in our brain so general that they can deal with either speech or vision, depending on what input it takes through the development? Why are robots that are programmed using human designed, task-specific representation do not do well in complex, changing, or partially unknown or totally unknown environment? What are the self-organization schemes that robots can use to autonomously develop their mental skills through interactions with the environment? Is it more advantageous to enable robots to autonomously

*An invited paper in Proc. 2nd Int’l Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems August 10-11, 2002 Edinburgh, Scotland

develop their mental skills than to program robots using human-specified, task-specific representation?

Although robot mental development is very much a new concept (Weng et al., 2000b), a lot of well-known self-organization tools can be used in designing a developmental robot. In this paper, we summarize our recent investigations on this new direction and hopefully provide some answers to the above questions. In the following sections, we first outline the previous and current projects related to robot mental development conducted in our group. Then a theory of autonomous mental development of robots is presented followed by the experimental results on the SAIL robot, a developmental robot constructed following this theory. A brief comparison to others' work is given before we draw the conclusion.

2. An outline of previous and current projects

Our decade-long effort in enabling a machine to grow its perceptual and behavioral capabilities has gone through four systems: Cresceptron (1991 - 1995), SHSOLIF (1993 - 2000), SAIL (1996 - present) and Dav (1999 - present).

Cresceptron is an interactive software system for visual recognition and segmentation (Weng et al., 1997). The major contribution is a method to automatically generate (grow) a network for recognition from training images. The topology of this network is a function of the content of the training images. Due to its general nature in representation and learning, it turned out to be one of the first systems that have been trained to recognize and segment complex objects of very different natures from natural, complex backgrounds. Although Cresceptron is a general developmental system, its efficiency is low.

SHOSLIF (Self-organizing Hierarchical Optimal Subspace Learning and Inference Framework) was the next project whose goal was to resolve the efficiency of self-organization. It automatically finds a set of Most Discriminating Features (MDF) using Principle Component Analysis (PCA) followed by Linear Discriminant Analysis (LDA), for better generalization. It is a hierarchical structure organized by a tree to reach a logarithmic time complexity. Using it in an observation-driven Markov Decision Process (ODMDP), SHOSLIF has successfully controlled the ROME robot to navigate in MSU's large Engineering Building in real-time using only video cameras, without using any range sensors (Weng and Chen, 1998). All the real-time computing was performed by a slow Sun SPARC Ultra-1 Workstation. Therefore, SHOSLIF is very efficient for real-time operation. However, it is not an incremental learning method.

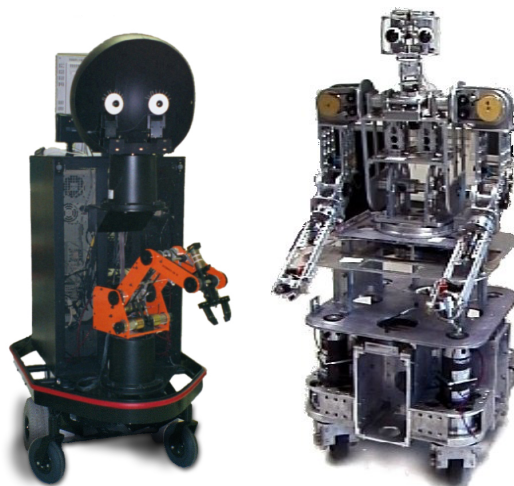


Figure 1: The SAIL (left) and Dav (right) robots.

SAIL (Self-organizing, Autonomous, Incremental Learner) is the next generation after SHOSLIF. The objective of this project is to automate the real-time incremental development for robot perceptual and behavioral capabilities. The internal representation of the SAIL robot (Fig. 1) is generated automatically by the robot itself, starting with a design of a coarse architecture. A self-organization engine called Incremental Hierarchical Discriminant Regression (IHDR) was the critical technology that achieves the stringent real-time, incremental, small sample size, large memory, and better generalization requirements (Hwang and Weng, 2000a) (Hwang and Weng, 2000b). IHDR automatically and incrementally grows and updates a tree (network) of nodes (remotely resemble cortical areas). In each node is an incrementally updated feature subspace, derived from the most discriminating features for better generalization. Discriminating features disregard factors that are not related to perception or actions, such as lighting in object recognition and autonomous navigation.

Dav robot (Fig. 1) is a humanoid robot, currently being developed as a next-generation test-bed for experimental investigations into autonomous mental development (Han et al., 2002). This general-purpose humanoid platform consists of a total of 43 degrees of freedom (DOF), including drive base, torso, arms, hands, neck and head. The body may support a wide array of locomotive and manipulative behaviors. For perception, Dav is equipped with a variety of sensing systems, including visual, auditory and haptic sensors. Its computational resource is totally onboard, including quadruple Pentium III plus PowerPCs, large memory and storage, networks, and long-sustenance power supply.

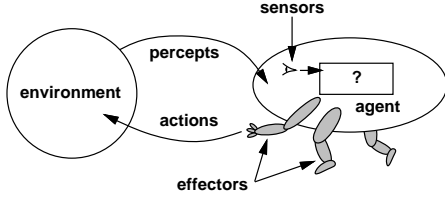


Figure 2: The abstract model of a traditional agent, which perceives the external environment and acts on it (adapted from (Russell and Norvig, 1995)). The source of perception and the target of action do not include the agent brain representation.

3. A theory for mentally developing robots

Evolving with the above robot projects is a theoretic framework for autonomous mental development of robots. We present the major components of this theory here. For more details, the reader is referred to (Weng, 2002).

3.1 SASE Agents

Defined in the standard AI literature (see, e.g., an excellent textbook (Russell and Norvig, 1995) and an excellent survey (Franklin and Graesser, 1996)), an agent is something that senses and acts, whose abstract model is shown in Fig. 2. As shown, the environment E of an agent is the world outside the agent.

To be precise in our further discussion, we need some mathematical notation. A context of an agent is a stochastic process (Papoulis, 1976), denoted by $g(t)$. It consists of two parts $g(t) = (x(t), a(t))$, where $x(t)$ denotes the sensory vector at time t which collects all signals (values) sensed by the sensors of the agent at time t , $a(t)$ the effector vector consisting of all the signals sent to the effectors of the agent at time t . The context of the agent from the time t_1 (when the agent is turned on) up to a later time t_2 is a *realization* of the random process $\{g(t) \mid t_1 \leq t \leq t_2\}$. Similarly, we call $\{x(t) \mid t_1 \leq t \leq t_2\}$ a sensory context and $\{a(t) \mid t_1 \leq t \leq t_2\}$ an action context.

The set of all the possible contexts of an environment E is called the context domain \mathcal{D} . As indicated by Fig. 2, at each time t , the agent senses vector $x(t)$ from the environment using its sensors and it sends $a(t)$ as action to its effectors. Typically, at any time t the agent uses only a subset of the history represented in the context, since only a subset is mostly related to the current action.

The model in Fig. 2 is for an agent that perceives only the external environment and acts on the external environment. Such agents range from a simple thermostat to a complex space shuttle. This well accepted model played an important role in agent re-

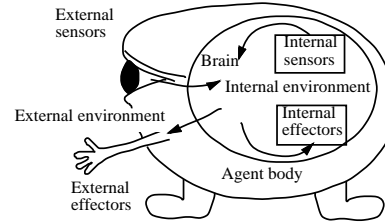


Figure 3: A self-aware self-effecting (SASE) agent. It interacts with not only the external environment but also its own internal (brain) environment: the representation of the brain itself.

search and applications. Unfortunately, this model has a fundamental flaw: It does not sense its internal “brain” activities. In other words, its internal decision process is neither a target of its own cognition nor a subject for the agent to explain.

The human brain allows the thinker to sense what he is thinking about without performing an overt action. For example, visual attention is a self-aware self-effecting internal action (see, e.g., (Kandel et al., 1991), pp. 396 - 403). Motivated by neuroscience, it is proposed here that a highly intelligent being must be *self-aware and self-effecting* (SASE). Fig. 3 shows an illustration of a SASE agent. A formal definition of a SASE agent is as follows:

Definition 1 *A self-aware and self-effecting (SASE) agent has internal sensors and internal effectors. In addition to interacting with the external environment, it senses some of its internal representation as a part of its perceptual process and it generates actions for its internal effectors as a part of its action process.*

Using this new agent model, the sensory context $x(t)$ of a SASE agent must contain information about not only external environment E , but also internal representation R . Further, the action context $a(t)$ of a SASE agent must include internal effectors that act on R .

A traditional non-SASE agent does use internal representation R to make decision. However, this decision process and the internal representation R is not included in what is to be sensed, perceived, recognized, discriminated, understood and explained by the agent itself. Thus, a non-SASE agent is not able to understand what it is doing, or in other words, it is not self-aware. Further, the behaviors that it generates are for the external world only, not for the brain itself. Thus, it is not able to autonomously change its internal decision steps either. For example, it is not able to modify its value system based on its experience about what is good and what is bad.

It is important to note that not all the internal brain representations are sensed by the brain itself. For example, we cannot sense why we have interesting visual illusions (Eagleman, 2001).

3.2 Autonomous mental development (AMD)

An agent can perform one, multiple or an open number of tasks. The task here is not restricted by type, scope, or level. Therefore, a task can be a subtask of another. For example, making a turn at a corner or navigating around a building can both be a task.

To enable an agent to perform certain tasks, the traditional paradigm involves developing task-specific architecture, representation, and skills through human hands, which we call it a “manual” development. The manual paradigm has two phases, the manual development phase and the automatic execution phase. In the first phase, a human developer H is given a specific task T to be performed by the machine and a set of ecological conditions E_c about operational environment. The human developer first understands the task. Next, he designs a task-specific architecture and representation and then programs the agent A . In mathematical notation, we consider a human as a (time varying) function that maps the given task T and the set of ecological conditions E_c to agent A :

$$A = H(E_c, T). \quad (1)$$

In the automatic execution phase, the machine is placed in the task-specific setting. It operates by sensing and acting. It may learn, using sensory data to change some of its internal parameters. However, it is the human who understands the task and programs the internal representation. The agent just runs the program.

Correspondingly, the autonomous development paradigm has two different phases, the construction and programming phase and the autonomous development phase.

In the first phase, tasks that the agent will end up learning are unknown to the robot programmer. The programmer might speculate some possible tasks, but writing a task-specific representation is not possible without actually given a task. The ecological conditions under which the robot will operate, e.g., land-based or underseas, are provided to the human developer so that he can design the agent body appropriately. He writes a task-nonspecific program called *developmental program*, which controls the process of mental development. Thus the newborn agent $A(t)$ is a function of a set of ecological conditions only, but not the task:

$$A(0) = H(E_c), \quad (2)$$

where we added the time variable t to the time varying agent $A(t)$, assuming that the birth time is at $t = 0$.

After the robot is turned on at time $t = 0$, the robot is “born” and it starts to interact with the

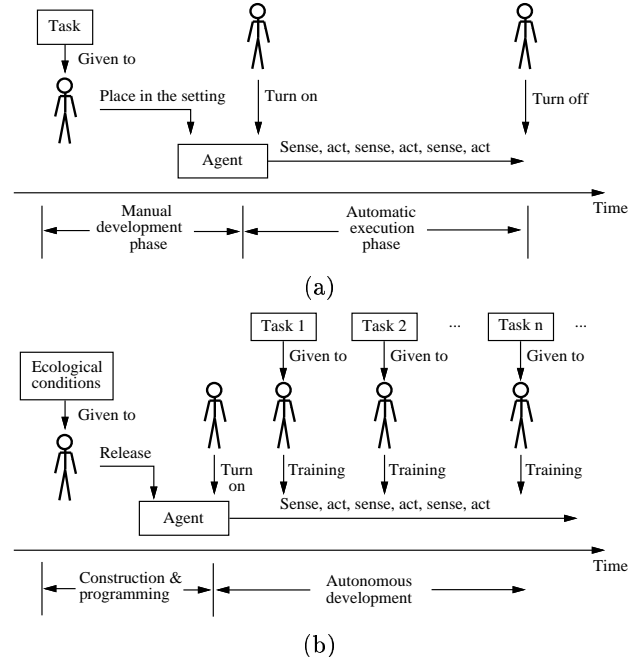


Figure 4: Manual development paradigm (a) and autonomous development (b) paradigm .

physical environment in real time by continuously sensing and acting. This phase is called autonomous development phase. Human teachers can affect the developing robot only as a part of the environment, through the robot’s sensors and effectors. After the birth, the internal representation is not accessible to the human teacher.

Various learning modes are available to the teacher during autonomous development. He can use supervised learning by directly manipulating (compliant) robot effectors (see, e.g., (Weng et al., 1999)), like how a teacher holds the hand of a child while teaching him to write. He can use reinforcement learning by letting the robot try on its own while the teacher encourages or discourages certain actions by pressing the “good” or “bad” buttons in the right context (see, e.g., (Weng et al., 2000a) (Zhang and Weng, 2001b)). The environment itself can also produce reward directly. For example, a “sweet” object and a “bitter” one (see, e.g., (Almassy et al., 1998)). With multiple tasks in mind, the human teacher figures out which learning mode is more suitable and efficient and he typically teaches one task at a time. Skills acquired early are used later by the robot to facilitate learning new tasks.

Fig. 4 illustrates the traditional manual development paradigm and the autonomous development paradigm.

3.3 Internal Representation

Autonomous generation of internal representation is central to mental development. Traditional AI systems use symbolic representation for internal representation and decision making. Is symbolic representation suited for a developmental robot? In the AI research, the issue of representation has not been sufficiently investigated, mainly due to the traditional manual development paradigm. There has been a confusion of concepts in representation, especially between reality and the observation made by the agents. To be precise, we first define some terms.

A *world concept* is a concept about objects in the external environment of the agent, which includes both the environment external to the robot and the physical body of the robot. The *mind concept*¹ is internal with respect to the nervous system (including the brain).

Definition 2 A world centered representation is such that every item in the representation corresponds to a world concept. A body centered representation is such that every item in the representation corresponds to a mind concept.

A mind concept is related to phenomena observable from the real world, but it does not necessarily reflect the reality correctly. It can be an illusion or totally false.

Definition 3 A symbolic representation is about a concept in the world and, thus, it is world centered. It is in the form $A = (v_1, v_2, \dots, v_n)$ where A (optional) is the name token of the object and v_1, v_2, \dots, v_n are the unique set of attributes of the object with predefined symbolic meanings.

For example, Apple = (weight,color) is a symbolic representation of a class of objects called apple. Apple-1 = (0.25g,red) is a symbolic representation of a concrete object called Apple-1. The set of attributes is unique in the sense that the object's weight is given by the unique entry v_1 . Of course, other attributes such as confidence of the weight can be used. A typical symbolic representation has the following characteristics:

1. Each component in the representation has a predefined meaning about the object in the external world.
2. Each attribute is represented by a unique variable in the representation.
3. The representation is unique for a single corresponding physical object in the external environment.

¹The term "mind" is used for ease of understanding. We do not claim that it is similar to the human mind.

World centered symbolic representation has been widely used in symbolic knowledge representation, databases, expert systems, and traditional AI systems.

Another type of representation is motivated by the distributed representation in the biological brain:

Definition 4 A distributed representation is not necessarily about any particular object in the environment. It is body centered, grown from the body's sensors and effectors. It is in a vector form $A = (v_1, v_2, \dots, v_n)$, where A (optional) denotes the vector and $v_i, i = 1, 2, \dots, n$ corresponds to either a sensory element (e.g., pixel or receptor) in the sensory input, a motor control terminal in the action output, or a function of them.

For example, suppose that an image produced by a digital camera is denoted by a column vector I , whose dimension is equal to the number of pixels in the digital image. Then I is a distributed representation, and so is $f(I)$ where f is any function. A distributed representation of dimension n can represent the response of n neurons.

The world centered and body centered representations are the same only in the trivial case where the entire external world is the only single object for cognition. There is no need to recognize different objects in the world. A thermostat is an example. The complex world around it is nothing more than a temperature to it. Since cognition must include discrimination, cognition itself is not needed in such a trivial case. Otherwise, body centered representation is very different from a world centered representation. Some later (later in processing steps) body centered representations can have a more focused correspondence to a world concept in a mature developmental robot, but they will never be identical. For example, the representation generated by a view of a red apple is distributed over many cortical areas and, thus, is not the same as a human designed atomic, world centered symbolic representation.

A developmental program is designed after the robot body has been designed. Thus, the sensors and effectors of the robot are known, and so are their signal formats. Therefore, the sensors and effectors are two major sources of information for generating distributed representation.

Another source of information is the internal sensors and effectors which may grow or die according to the autonomously generated or deleted representation. Examples of internal effectors include attention effectors in a sensory cortex and rehearsal effectors in a premotor cortex. An internal attention effectors are used for turning on or turning off certain signal lines for, e.g., internal visual attention. Rehearsal effectors are useful for planning before an action is actually released to the motors. The internal sensors include those that sense internal effectors. In

fact, all the conscious internal effectors should have corresponding internal sensors.

It seems that a developmental program should use a distributed representation, because the tasks are unknown at the robot programming time. It is natural that the representation in earlier processing is very much sensor centered and the representation in later processing is very much effector centered. Learned associations map perceptually very different sensory inputs to the same equivalent class of actions. This is because a developmental being is shaped by the environment to produce such a desired behavior.

On the other hand, an effector centered representation can correspond to a world object well. For example, when the eyes of a child sense (see) his father’s portrait and his ears sense (hear) a question “who is he?” The internally primed action can be any of the following actions: saying “he is my father,” “my dad,” “my daddy,” etc. In this example, the later action representation can correspond to a world object, “father,” but it is still a (body centered) distributed representation. Further, since the generated actions are not unique given different sensory inputs of the same object, there is no place for the brain (human or robot) to arrive at a unique representation from a wide variety of sensory contexts that reflects the world that contains the same single object as well as others. For example, there is no way for the brain to arrive at a unique representation in the above “father” example. Therefore, a symbolic representation is not suited for a developmental program while a distributed representation is.

4. SAIL - An example of developmental robots

The SAIL robot is our current autonomous developmental process test-bed. It is a human-size mobile robot house-made at Michigan State University with a drive-base, a six-joint robot arm, a rotary neck, and two pan-tilt units, on which two CCD cameras (as eyes) are mounted. A wireless microphone functions as an ear. The SAIL robot has four pressure sensors on its torso and 28 touch sensors on its eyes, arm, neck, and bumper. Its main computer is a dual-processor dual-bus PC workstation with 512 MB RAM and a 27 GB three-drive disk array. All the sensory information processing, memory recall and update as well as real-time effector controls are done in real-time.

According to the theory presented in Section 3., our SAIL developmental algorithm has some “innate” reflexive behaviors built-in. At the “birth” time of the SAIL robot, its developmental algorithm starts to run. This developmental algorithm runs in real time, through the entire “life span” of the robot. In other words, the design of the developmental pro-

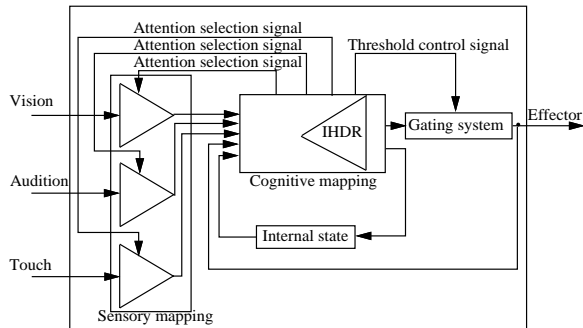


Figure 5: Partial internal architecture of a single level in the SAIL developmental program.

gram cannot be changed once the robot is “born,” no matter what tasks that it ends up learning. The robot learns while performing simultaneously. The innate reflexive behaviors enable it to explore the environment while improving its skills. The human trainers train the robot by interacting with it, very much like the way human parents interact with their infant, letting it seeing around, demonstrating how to reaching objects, teaching commands with the required responses, delivering reward or punishment (pressing “good” or “bad” buttons on the robot), etc. The SAIL developmental algorithm updates the robot memory in real-time according to what was sensed by the sensors, what it did, and what it received as feedback from the human trainers.

4.1 Architecture

The schematic architecture of a single level of SAIL is shown in Fig. 5. Sensory inputs first enter a module called sensory mapping, whose detailed structure is discussed in Section 4.2.

Internal attention for vision, audition and touch, is a very important mechanism for the success of multimodal sensing. A major challenge of perception for high dimensional data inputs such as vision, audition and touch is that often not all the lines in the input are related to the task at hand. Attention selection enables singles of only a bundle of relevant lines are selected for passing through while others are blocked. Attention selection is an internal effector since it acts on the internal structure of the “brain” instead of the external environment.

First, each sensing modality, vision, audition and touch, needs intra-modal attention to select a subset of internal output lines for further processing but disregard to leaving unrelated other lines. Second, the inter-modal attention, which selects a single or multiple modalities for attention. Attention is necessary because not only do our processors have only a limited computational power, but more importantly, focusing on only related inputs enables powerful generalization.

The cognitive mapping module is the central part of the system. It is responsible for learning the association between the sensory information, the context, and the behavior. The behaviors can be both external and internal. The external behaviors correspond to control signals for external effectors such as the joint motors of a robot arm, or whatever peripherals that the robot has to act on the environment. The internal behaviors include the above-mentioned attention selection signals for the sensory mapping module, the effector that manipulates the internal states and the threshold control signals to the gating system. The cognitive mapping is implemented by the IHDR tree, which mathematically computes the mapping,

$$g : S \times X \rightarrow S \times A,$$

where S is the state (context) space, X is the sensory space, and A is the action space. IHDR derives the best features that are most relevant to output by doing a double clustering in both input and output space. It constructs a tree structure and repeats the double clustering in a coarse-to-fine manner in each of the tree nodes. The resulted tree structure is used to find the best matching input cluster in a fast logarithmic time. Compared to other methods, such as artificial neural network, linear discriminant analysis, and principal component analysis, IHDR has advantages in handling high-dimensional input, doing discriminant feature selection, and learning from one instance.

The gating system evaluates whether the intended action accumulates sufficient thrust to be issued as an actual action. In this way, actions are actually made only when a sufficient number of action primitives are given through the time by the cognitive mapping module. This mechanism significantly reduces the requirement on the accuracy of timing of issued action primitives.

Three types of learning modes have been implemented on SAIL: learning by imitation (supervised learning), reinforcement learning, and communicative learning. In the following sections, we explain how learning is conducted by the SAIL robot while providing the experimental results.

4.2 Staggered Hierarchical Mapping

We have designed and implemented a sensory mapping method, called “Staggered Hierarchical Mapping (SHM),” shown in Fig. 6, and its developmental algorithm (Zhang and Weng, 2002a). Its goal includes: (1) to generate feature representation for receptive fields at different positions in the sensory space and with different sizes and (2) to allow attention selection for local processing. SHM is a model motivated by human early visual pathways including processing performed by the retina, Lat-

eral Geniculate Nucleus (LGN) and the primary visual cortex. A new Incremental Principal Component Analysis (IPCA) method is used to automatically develop orientation sensitive and other needed filters (Zhang and Weng, 2001a). From sequentially sensed video frames, the proposed algorithm develops a hierarchy of filters, whose outputs are uncorrelated within each layer, but with increasing scale of receptive fields from low to high layers. To study the completeness of the representation generated by the SHM, we experimentally showed that the response produced at any layer is sufficient to reconstruct the corresponding “retinal” image to a great degree. This result indicates that the internal representation generated for receptive fields at different locations and sizes are nearly complete in the sense that it does not lose important information. The attention selection effector is internal and thus cannot be guided from the “outside” by a human teacher. The behaviors for internal effectors can be learned through reinforcement learning and communicative learning.

4.3 Vision-guided navigation

In the experiment of vision-guided navigation (Weng et al., 2000a), a human teacher teaches the robot by taking it for a walk along the hallways of MSU Engineering Building. Force sensors on the robot body sense the push action of the teacher and its two drive wheels complies by moving at a speed that is proportional to the force that is sensed each side. In other words, the robot performs supervised learning in real time through imitation.

The IHDR mapping algorithm processes the input image in real time. It derives features that are related to the action but disregard features that are not. The human teacher does not need to define features. The system runs at about 10 Hz, 10 updates of navigation decisions per second. In other words, for each 100 millisecond, a different set of feature subspaces are used. To address the requirement of real-time speed, the IHDR method incrementally constructs a tree architecture which automatically generates and updates the representations in a coarse to fine fashion. The real-time speed is achieved by the logarithmic time complexity of the tree in that the time required to update the tree for each sensory frame is a logarithmic function in the number of fine clusters (prototypes) in the tree.

After 4 trips along slightly different trajectories along the hallways, the human teacher started to let the robot go free. He needed to “hand push” the robot at certain places when necessary until the robot could reliably navigate along the hallway, without a need for “hand-lead.” We found that about 10 trips were sufficient for the SAIL robot to navigate along the hallways, using only vision, without using

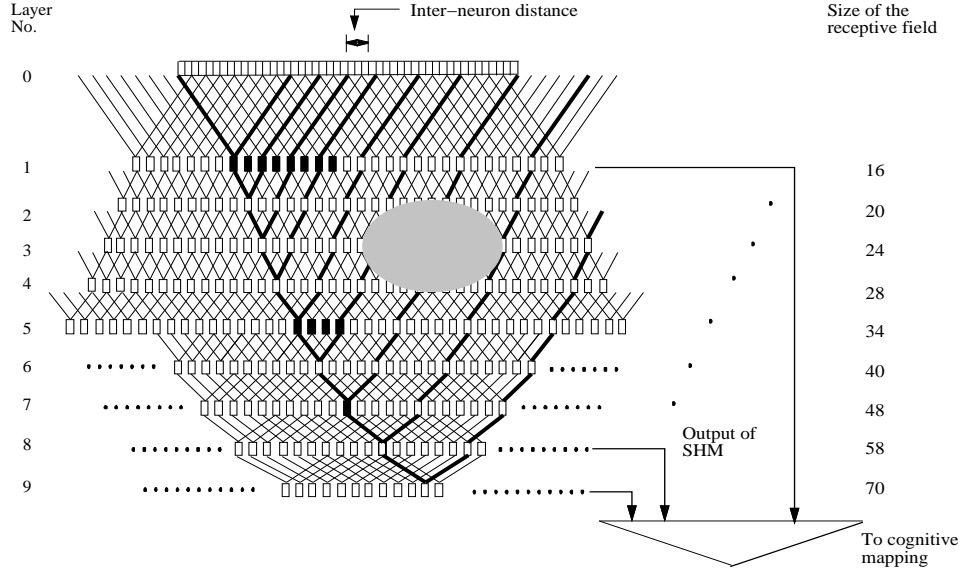


Figure 6: The architecture of SHM. Each square denotes a neuron. Layer 0 is the input image. The neurons marked as black in layer 1 belong to the same eigen-group. Bold lines that are derived from a single neuron and expanded to the original image mark the receptive field of that neuron. The size of the receptive field in a particular layer is 20% larger than its previous layer in this diagram, which is shown at the right. The size of the receptive field is rounded to the nearest integer. SHM allows not only a bottom up response computation, but also a top down attention selection. The oval indicates the lines selected by attention selector.

any range sensors. Fig. 7 shows some images that the robot saw during the navigation.

4.4 Grounded speech learning

Similar to learning vision-guided navigation, the SAIL robot can learn to follow voice command through physical interaction with a human trainer (Zhang and Weng, 2001b). In the early supervised learning stage, a trainer spoke a command to the robot and then executed a desired action by pressing a pressure sensor or a touch sensor that was linked to the corresponding effector. At later stages, when the robot can explore more or less on its own, the human teacher uses reinforcement learning by pressing its “good” or “bad” button to encourage and discourage certain actions. Typically, after about 15-30-minute interactions with a particular human trainer, the SAIL robot could follow commands with about 90% correct rate. Table 1 shows the voice commands learned by the SAIL robot and its performance. Fig. 8 shows the graphic user interface for humans to monitor the progress of online grounded speech learning.

4.5 Communicative learning

Recently, we have successfully implemented the new communicative learning mode on the SAIL robot. First, in the language acquisition stage, we taught SAIL simple verbal commands, such as “go ahead,”

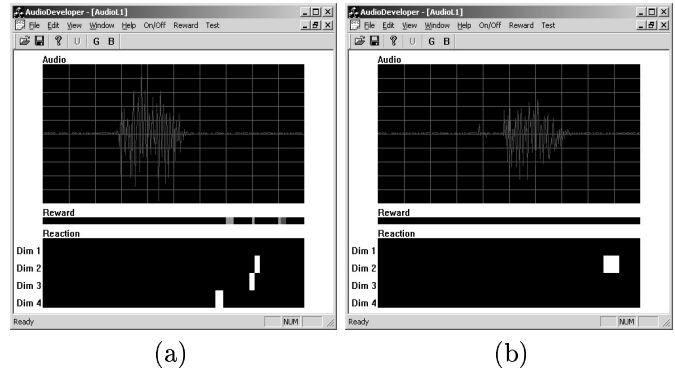


Figure 8: The GUI of AudioDeveloper: (a) During online learning; (b) After online learning.

“turn left,” “turn right,” “stop,” “look ahead,” “look left,” “look right,” etc by speaking to it online while guiding the robot to perform the corresponding action. In the next stage, teaching using language, we taught the SAIL robot what to do in the corresponding context through verbal commands. For example, when we wanted the robot to turn left (a fixed amount of heading increment), we told it to “turn left.” When we wanted it to look left (also a fixed amount of increment), we told it to “look left.” This way, we did not need to physically touch the robot during training and used instead much more sophisticated verbal commands. This made training more efficient and more precise. Fig. 9 shows the SAIL robot navigating in real-time along the corridors of

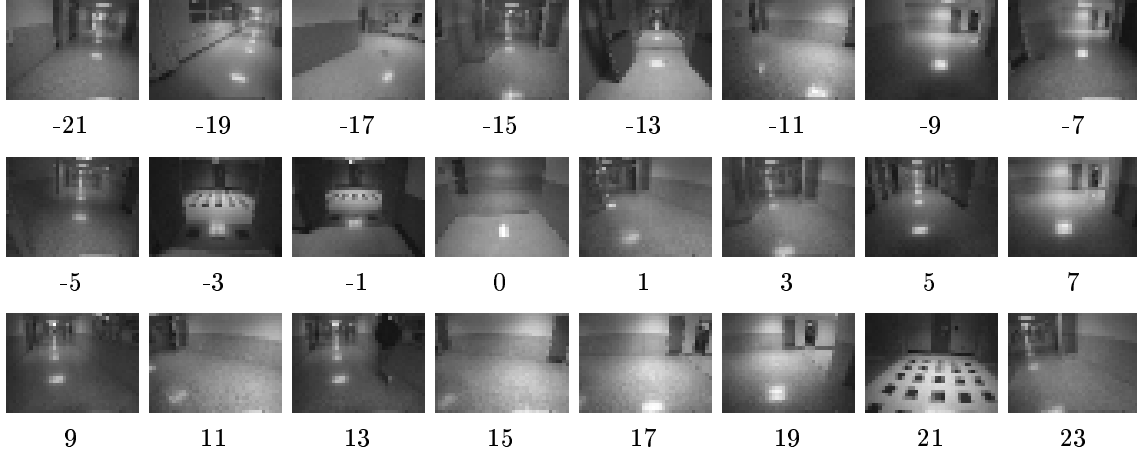


Figure 7: A subset of images used in autonomous navigation problem. The number right below the image shows the needed heading direction (in degrees) associated with that image.

Table 1: Performance of the SAIL robot in grounded speech learning. After training, the trainer tested the SAIL robot by guiding it through the second floor of Engineering Building. As SAIL did not have perfect heading alignment, the human trainer used verbal commands to adjust robot heading during turns and straight navigation. During the navigation, the arm and eye commands are issued 10 times each at different locations.

Commands	Go left	Go right	Forward	Backward	Freeze
Correct rate(%)	88.9	89.3	92.8	87.5	88.9
Commands	Arm left	Arm right	Arm up	Arm down	Hand open
Correct rate(%)	90	90	100	100	90
Commands	Hand close	See left	See right	See up	See down
Correct rate(%)	80	100	100	100	100

the Engineering Building, at a typical human walking speed.

4.6 Action chaining

The capability of learning new skills is very important for an artificial agent to scale up. We have designed and implemented a hierarchical developmental learning architecture (Fig. 10), which enables a robot to develop complex behaviors (chained actions) after acquisition of simple ones (Zhang and Weng, 2002b). The mechanism that makes this possible is chained secondary conditioning. An action chaining process can be written mathematically as,

$$C_c \rightarrow C_{s1} \rightarrow A_{s1} \rightarrow C_{s2} \rightarrow A_{s2} \Rightarrow C_c \rightarrow A_{s1} \rightarrow A_{s2} \quad (3)$$

where C_c is the composite command, C_{s1} and C_{s2} are commands invoking basic actions A_{s1} and A_{s2} , respectively. \rightarrow means “followed by”, and \Rightarrow means “develops”. The problem here is that C_{s1} and C_{s2} are missing in the developed stimuli-response association. The major challenge of this work is that training and testing must be conducted in the same mode through online real-time interactions between the robot and the trainer.

In the experiment, upon learning the basic gripper

tip movements (Fig. 11), the SAIL robot learned to combine individually instructed movements to be a composite one invoked by a single verbal command without any reprogramming (Fig. 12). To solve the problem of missing context in action chaining, we modeled a primed context as the follow-up sensation and action of a real context. By backpropagating the primed context, a real context was able to predict future contexts, which enabled the agent to react correctly even with some missing contexts. The learning strategy integrated supervised learning and reinforcement learning. To handle the “abstraction” issue in real sensory inputs, a multi-level architecture was used with the higher level emulating the function of higher-order cortex in biology in some sense.

5. Value system

A value system of a robot enables the robot to know what is bad and what is good, and to act for the good. Without a value system, a robot either does nothing or does every move mechanically and thus lacks intelligence. We have designed and implemented a low level value system for the SAIL robot. The value system integrates the habituation mechanism and reinforcement learning so that the robot’s responses to certain visual stimuli would change after



Figure 9: SAIL robot navigates autonomously using its autonomously developed visual perceptual behaviors. Four movies are available at <http://www.egr.msu.edu/mars/> to provide more results.

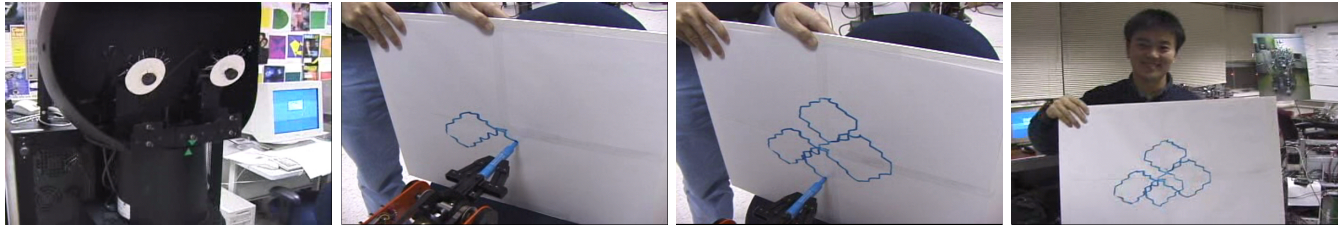


Figure 12: The SAIL robot learned the chained action after verbally instructed by human trainers.

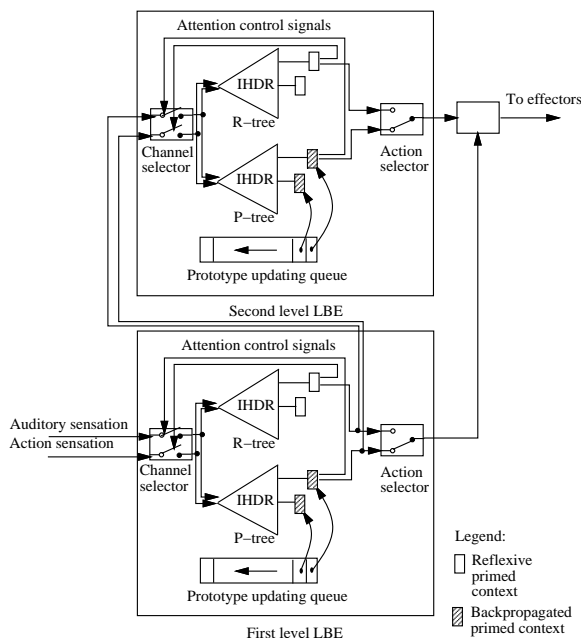


Figure 10: A hierarchical developmental learning architecture for action chaining.

interacting with human trainers. For more details, the reader is referred to another paper in the proceeding of this workshop (Huang and Weng, 2002).

6. Comparison with others' work

What is the most basic difference between a traditional learning algorithm and a developmental algorithm? Autonomous development does require a capability of learning but it requires something more fundamental. A developmental algorithm must be

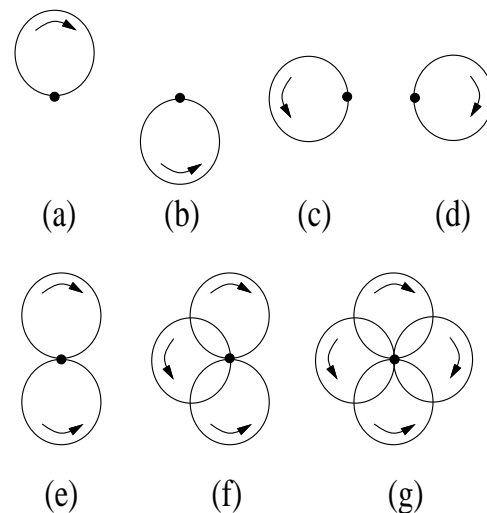


Figure 11: Gripper tip trajectories of the SAIL robot. (a)-(d) are basic actions, each of which starts from the black dot. (e)-(g) are composite actions chaining some or all of the basic ones.

able to learn tasks that its programmers do not know or even cannot predict. This is because a developmental algorithm, once designed before robot "birth," must be able to learn new tasks and new skills without requiring re-programming. The representation of a traditional learning algorithm is designed by humans for a given task but that for a developmental algorithm must be autonomously generated. As a working example, humans' developmental algorithm enables humans to learn new skills without a need to change the design of their brain.

However, the motive of developmental robots is not to make robot more difficult to program, but rel-

atively easier instead. The task nonspecific nature of a developmental program is a blessing. It relieves human programmers from the daunting tasks of programming task-specific visual recognition, speech recognition, autonomous navigation, object manipulation, etc, for unknown environments. The programming task for a developmental algorithm concentrates on self-organization schemes, which are more manageable by human programmers than the above task-specific programming tasks for unknown or partially unknown environments.

Designing and implementing a developmental program are systematic, clearly understandable using mathematical tools. Designing a perception program and its representation in a task-specific way using a traditional approach, however, is typically very complex, ad hoc, and labor intensive. The resulting system tends to be brittle. Design and implementation of a developmental program are of course not easy. However, the new developmental approach is significantly more tractable than the traditional approaches in programming a perception machine. Further, it is applicable to uncontrolled real-world environments, the only approach that is capable of doing this.

Due to its cross-environment capability, the SAIL robot has demonstrated vision-guided autonomous navigation capability in both complex outdoor and indoor environments. The Hierarchical Discriminant Regression (HDR) engine played a central role in this success. Although ALVINN at CMU (Pomerleau, 1989) can in principle be applied to indoor, however the local minima and loss of memory problem with artificial intelligence make it very difficult to work in the complex indoor scenes.

The SAIL robot has successfully developed real-time, integrated multimodal (vision, audition, touch, keyboard and via wireless network) human-robot interaction capability, to allow a human operator to enter different degrees of intervention seamlessly. A basic reason for achieving this extremely challenging capability is that the SAIL robot is developed to associate over tens of thousands of multi-modal contexts in real-time in a grounded fashion, which is another central idea of AMD. Some behavior-based robots such as Cog and Kismet at MIT do online interactions with humans, but they are off-line hand programmed. They cannot interact with humans while learning.

The perception-based action chaining allows the SAIL robot to develop complex perception-action sequences (or behaviors) from simple perception-action sequences (behaviors) through real-time online human robot interactions, all are done in the same continuous operational mode. This capability appears simpler than it really is. The robot must infer about context in high-dimensional perception

vector space. It generates new internal representation and uses it for later context prediction, which is central for scaling up in AMD. David Touresky's skinnerbot (Touretzky and Saksida, 1999) does action chaining, but it does it through preprogrammed symbols and thus the robot is not applicable to unknown environments.

7. Conclusion

For a robot, every action is context dependent, i.e., it is tightly dependent on the rich information available in the sensory input and the state. The complexity of the rules of such context dependence is beyond human programming, which is one of the fundamental reasons that traditional ways have been proved to be extremely difficult to develop robots running in a typical human environment.

We introduced here a new kind of robots – developmental robots that can develop their mental skills automatically through real-time interactions with the environment. Motivated by human mental development from infancy to adulthood, the proposed theoretical framework have been proved on the SAIL robot in multiple tasks, from vision-guided navigation, grounded speech learning, to behavior scale-up through action chaining, all learned and performed online in real time. The main reason behind this achievement is that the robot does not rely on human to pre-define representation. The representation of the system is automatically generated through the interaction between the developmental mechanism and the experience. We believe what we have achieved is a starting point of the promising new direction of robotics. While there are yet plenty of practical questions waiting for us to answer, it opens a wide range of opportunities for future research.

Acknowledgements

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, DARPA ITO under grant No. DABT63-99-1-0014, an MSU Strategic Partnership Grant, and a research gift from Microsoft Research and Zyvex. Many thanks to M. Badgero, C. Evens, J.D. Han, W.S. Hwang, X. Huang, K.Y. Tham, S.Q. Zeng and N. Zhang for their contributions to the SAIL and DAV projects as referred.

References

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

- Eagleman, D. M. (2001). Visual illusions and neurobiology. *Nature Reviews: Neuroscience*, 2(2):920–926.
- Franklin, S. and Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proc. the 3rd Int'l Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag.
- Han, J., Zeng, S., Tham, K., Badgero, M., and Weng, J. (2002). Dav: A humanoid robot platform for autonomous mental development. In *Proc. IEEE 2nd International Conference on Development and Learning (ICDL 2002)*, MIT, Cambridge, MA.
- Huang, X. and Weng, J. (2002). Novelty and reinforcement learning in the value system of developmental robots. In *Proc. Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems (EPIROB 2002)*, Edinburgh, Scotland.
- Hwang, W. and Weng, J. (2000a). Hierarchical discriminant regression. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1277–1293.
- Hwang, W. and Weng, J. (2000b). An online training and online testing algorithm for OCR and image orientation classification using hierarchical discriminant regression. In *Proc. Fourth IAPR International Workshop on Document Analysis Systems*, Rio De Janeiro, Brazil.
- Kandel, E. R., Schwartz, J. H., and Jessell, T. M., (Eds.) (1991). *Principles of Neural Science*. Appleton and Lance, Norwalk, Connecticut, third edition.
- Papoulis, A. (1976). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, NY, 2nd edition.
- Pomerleau, D. A. (1989). Alvin: An autonomous land vehicle in a neural network. In Touretzky, D., (Ed.), *Advances in Neural Information Processing*, volume 1, pages 305–313. Morgan-Kaufmann Publishers, San Mateo, CA.
- Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Upper Saddle River, NJ.
- 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.
- Touretzky, D. and Saksida, L. (1999). Operant conditioning in skinnerbots. *Adaptive Behavior*, 5(3/4):219–247.
- von Melchner, L., Pallas, S. L., and Sur, M. (2000). Visual behavior induced by retinal projections directed to the auditory pathway. *Nature*, 404:871–875.
- Weng, J. (2002). A theory for mentally developing robots. In *Proc. IEEE 2nd International Conference on Development and Learning (ICDL 2002)*, MIT, Cambridge, MA.
- Weng, J., Ahuja, N., and Huang, T. S. (1997). Learning recognition using the Cresceptron. *Int'l Journal of Computer Vision*, 25(2):109–143.
- Weng, J. and Chen, S. (1998). Vision-guided navigation using SHOSLIF. *Neural Networks*, 11:1511–1529.
- Weng, J., Hwang, W., and Zhang, Y. (2000a). Developmental humanoids: humanoids that develop skills automatically. In *Proc. The First IEEE-RAS International Conference on Humanoid Robots*, Boston, MA.
- Weng, J., Hwang, W. S., Zhang, Y., and Evans, C. (1999). Developmental robots: Theory, method and experimental results. In *Proc. 2nd International Conference on Humanoid Robots*, pages 57–64, Tokyo, Japan.
- Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2000b). Autonomous mental development by robots and animals. *Science*, 291:599–600.
- Zhang, N. and Weng, J. (2002a). A developing sensory mapping for robots. In *Proc. IEEE 2nd International Conference on Development and Learning (ICDL 2002)*, MIT, Cambridge, MA.
- Zhang, Y. and Weng, J. (2001a). Complementary candid incremental principal component analysis. Technical Report MSU-CSE-01-24, Department of Computer Science and Engineering, Michigan State University, East Lansing, MI.
- Zhang, Y. and Weng, J. (2001b). Grounded auditory development of a developmental robot. In *Proc. INNS-IEEE International Joint Conference on Neural Networks*, pages 1059–1064, Washington, DC.
- Zhang, Y. and Weng, J. (2002b). Action chaining by a developmental robot with a value system. In *Proc. IEEE 2nd International Conference on Development and Learning (ICDL 2002)*, MIT, Cambridge, MA.