

DEVELOPMENTAL ROBOTICS: THEORY AND EXPERIMENTS

JUYANG WENG

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

Received 17 October 2003

Revised 23 February 2004

Accepted 22 March 2004

A hand-designed internal representation of the world cannot deal with unknown or uncontrolled environments. Motivated by human cognitive and behavioral development, this paper presents a theory, an architecture, and some experimental results for developmental robotics. By a developmental robot, we mean that the robot generates its “brain” (or “central nervous system,” including the information processor and controller) through online, real-time interactions with its environment (including humans). A new Self-Aware Self-Effecting (SASE) agent concept is proposed, based on our SAIL and Dav developmental robots. The manual and autonomous development paradigms are formulated along with a theory of representation suited for autonomous development. Unlike traditional robot learning, the tasks that a developmental robot ends up learning are unknown during the programming time so that the task-specific representation must be generated and updated through real-time “living” experiences. Experimental results with SAIL and Dav developmental robots are presented, including visual attention selection, autonomous navigation, developmental speech learning, range-based obstacle avoidance, and scaffolding through transfer and chaining.

Keywords: Cognitive development; agents; AI architecture, controller architecture; representation; human machine interfaces; attention selection; autonomous navigation; obstacle avoidance; speech recognition; object recognition; scaffolding; transfer and chaining.

1. Introduction

In his pioneering paper published in 1950 titled “Computing Machinery and Intelligence,”⁴⁶ Alan Turing envisioned a machine that can learn like a child, which he called “child machine.” Due to a severe lack of computer-controlled machinery at that time, Turing suggested in that paper a disembodied abstract machine and proposed an “imitation game,” now called the Turing Test, to test machine intelligence.

Not until the 1980s had the importance of embodiment received sufficient recognition in the AI community. The behavior-based approach, popularized by

Rodney Brooks⁷ and others,³ put situated embodiment back on the AI stage as it deserves.

However, robot autonomous *mental*^a development did not receive sufficient attention, until the late 1990s when the SAIL^b robot^{51,55} and the Darwin V robot¹ started experiments on autonomous cognitive development. A 2001 article⁵⁷ in *Science* summarized the pivotal role that autonomous mental development (AMD) should play in both AI and our understanding of natural intelligence.

Traditional research paradigms in machine learning have been fruitfully informed by models of human learning. However, existing behavior-based learning techniques that are typically applied to robot learning²⁰ differ fundamentally from human mental development. For example, a task-specific representation is designed by the human programmer and only hand-designed parameters are learned by the machine. This greatly limits the power of machine capabilities in dynamic uncontrolled environments, such as vision, audition and language understanding. In contrast, a human child can learn concepts that none of his ancestors knew about (e.g. the concept of the Internet). Thus, it is unlikely that the representation of these new concepts (e.g. the Internet) are pre-designed by the genes.

This and other many differences are still not widely understood. Further, there is a need for basic theoretic frameworks for the new paradigm of autonomous mental development.

This article takes up some basic theoretical issues and describes the developmental robots SAIL and Dav^c that implement the theory. It does not describe algorithmic details but provides references to our prior publications where these details are available. We first introduce a new kind of agent, the Self-Aware Self-Effecting (SASE) agent, for autonomous mental development. Section 3 presents the paradigm of autonomous mental development (AMD). Section 4 introduces the software architecture of the SAIL and Dav developmental robots. Section 5 deals with the issue of representation, and argues for the inapplicability of symbolic representation to mental development. Section 6 briefly describes some experimental results with the SAIL and Dav developmental robots, which support the theory. Section 7 discusses some other related experimental studies. Section 8 provides concluding remarks.

2. SASE Agents

Defined in the standard AI literature (see, e.g. an excellent text by Russell and Norvig³⁹ and a survey by Franklin¹⁶), an agent is something that senses and acts, whose abstract model is shown in Fig. 1. As shown, the environment E of an agent is the world outside the agent.

^aThe term “mind” is used for a developmental robot, but we do not claim that the mind of a developmental robot is similar to a biological one.

^bStands for Self-Organizing Autonomous Incremental Learner.

^cA variant of “development.”

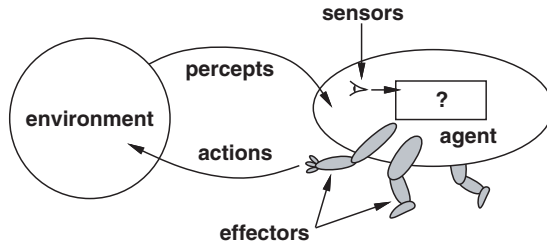


Fig. 1. The abstract model of a traditional agent, which perceives the external environment and acts on it (adapted from Ref. 39). The source of perception and the target of action do not include the agent’s brain representation.

A context $c(t)$ of an agent is a stochastic process.^d It consists of two parts $c(t) = (x(t), a(t))$, 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 , and $a(t)$ denotes the effector vector consisting of all the signals sent to the effectors at time t . The context of the agent from the previous time t_1 (after the agent is turned on) up to a later time t_2 is a *realization* of the stochastic process $\{c(\tau) \mid t_1 \leq \tau \leq t_2\}$. Typically, at any time t the agent uses only a subset of the context $c(t)$, since only a subset is most related to the required cognition and behavior.

Definition 1. The *internal environment* of an agent is the “brain” (or “the central nervous system”) of the agent. The *external environment* consists of all the remaining parts of the world, including the agent’s own body (excluding the brain).

The model in Fig. 1 is for an agent that perceives only the external environment and only acts on the external environment. Such agents range from a simple thermostat to a complex space shuttle. This well accepted model has played an important role in agent research 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 target for it to modify.

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 and self-effecting internal action (see, e.g. Ref. 24, pp. 396–403). Motivated by neuroscience, it is proposed here that a highly intelligent being must be *self-aware and self-effecting* (SASE), as shown in Fig. 2.

Definition 2. A *self-aware and self-effecting (SASE) agent* has internal sensors (IS) and internal effectors (IE) for the internal environment (brain), in addition to external sensors (ES) and external effectors (EE) for external environment (outside

^dA stochastic process is a series of random numbers or vectors $c(t)$, where for each fixed t , $c(t)$ is a random variable or vector.

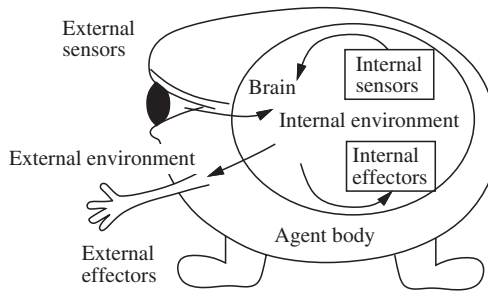


Fig. 2. A self-aware self-effecting (SASE) agent. It interacts with not only the external environment but also its own internal (brain) environment: the representation of the brain itself.

the brain). Both the internal and external environments are used (via IS and ES, respectively) as context of perception and cognition and the result of such perception and cognition is used to generate internal and external actions (via IE and EE, respectively). In order to be aware of a task in the (internal and external) environment, the agent experiences the distribution of contexts in the environment, learns to take alternative actions and memorizes their associated effects. The distribution of contexts and actions is across the environment (i.e. both related and unrelated to the task).

For example, attention selection and action release are internal actions and the senses of these actions are internal senses. Internally sensing what could be done (planning without actually doing) is internal sensing and deciding whether it is good to do now is an internal action (releasing the planned action to the effector).

A traditional non-SASE agent does use internal representation R to make decisions. 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 self-aware of its internal decision rules. Further, the behaviors that it generates are for the external world only, not for the brain itself. Thus, it is not able to modify its programmed-in, task-specific decision rules based on its new experience about what is good and what is bad.

Without experiencing contexts beyond those related to a task, the agent is not able to “step back” (from what it does) to examine and improve what it does. It is important to note that not all the internal brain representations are sensed by the brain itself (e.g. we have interesting visual illusions).

3. Machine Development Paradigms

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 around a corner or navigating around a building both can be a task.

3.1. Manual development

The term “manual” refers to developing task-specific architecture, representation and skills by human hands. The manual paradigm has two phases, the manual development phase and the automatic execution phase (see Fig. 3(a)). 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 the operational environment. The human developer first understands the task. Next, he designs a task-specific architecture and representation and then programs the agent A . If the human cannot determine all the parameters of his designed representation, he may use traditional machine learning during which he uses the sensory data to determine the parameters. 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 :

$$A = H(E_c, T). \tag{1}$$

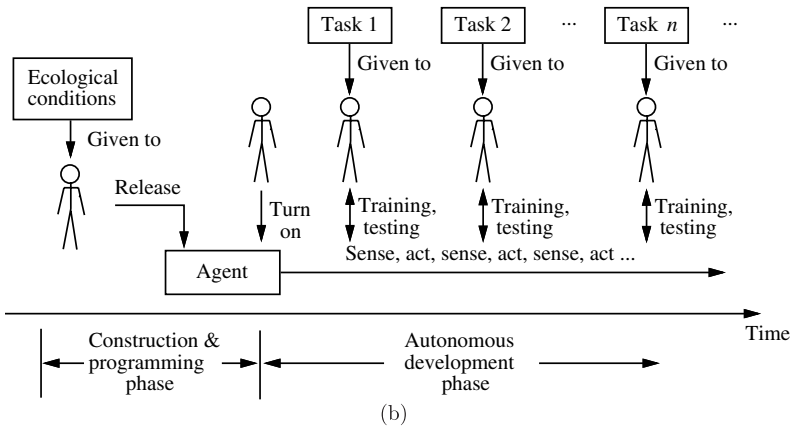
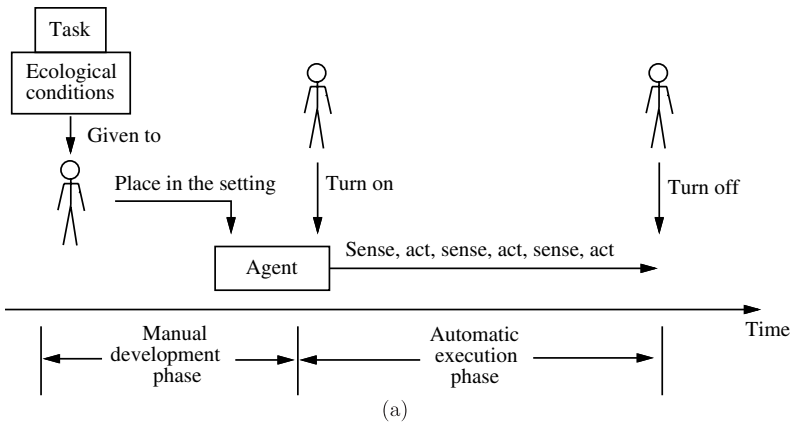


Fig. 3. (a) Manual development paradigm and (b) autonomous development paradigm.

In the second (automatic execution) phase, the machine is placed in a similar task-specific setting. It operates by sensing and acting. During this phase, the traditional machine learning may be conducted which further changes the human designed parameters using sensory data.

3.2. *Autonomous development*

The autonomous development paradigm has two phases, first the construction and programming phase and second the autonomous development phase (see Fig. 3(b)).

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 being given a task. The ecological conditions E_c under which the robot will operate, e.g., land-based or underwater, are provided to the human constructor so that he can design the agent body (e.g. sensors and effectors) appropriately. However, the given ecological conditions E_c are not useful for internal representation (since the specific environments are unknown and unpredictable). The human programmer writes a task-nonspecific program called a *developmental program*, which controls the process of autonomous 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 consider the agent to be time varying $A(t)$ with birth time $t = 0$.

After the robot is turned on at time $t = 0$, the robot is “born” and starts to interact with the physical environment in real-time by continuously sensing and acting. 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 teachers.

Various learning modes are available to a teacher during autonomous development. He can use supervised learning by directly manipulating (compliant) robot effectors,⁵⁵ like how a teacher holds the hand of a child while teaching him to draw a shape. 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” button in the right context^{56,67}. The environment itself can also produce rewards directly (e.g. “sweet” or “bitter” objects¹).

A more powerful learning mode, communicative learning, is introduced here:

Definition 3. *Communicative learning* is a type of learning mode that requires two processes, which can be interleaved through development: (i) Grounded language acquisition (using any mode of learning) and (ii) teaching using the acquired language.

Since the language is acquired through grounded experience, the meaning of the language is directly linked to physical senses and actual actions. Depending on the

sophistication of the meaning of the language used (words, phrases, or a full natural language), teaching using language is often significantly more effective than both supervised learning and reinforcement. For example, we will see in Sec. 6.6 how the SAIL robot acquired grounded meaning of each verbal command first and then used acquired language (verbal commands) to learn how to draw a flower in a new task setting. Of course, the learner also plays an active role in learning.⁴²

3.3. Relation between SASE and AMD

We consider two issues: (i) the capability to try alternatives (e.g. task rules) and (ii) the number of alternatives.

First, if a task-specific internal rule is hand-designed as in a manual development paradigm, the agent does not have the option and capability of constructing the rules autonomously and, thus, it is unable to understand the rules. For example, if a program runs only a hand-designed binary search algorithm without a chance to compare other search algorithms, it does not understand the pros and cons of binary search compared with other search algorithms. Only if the agent has the experience of autonomously *constructing* and trying (or being told about) various alternative internal rules according to the internal and external contexts (sensing experience) and *sensing* (or being told about) the effects of such a construction (effecting experience), can it understand the involved internal rules. In other words, *no alternatives, no understanding*. Without understanding, an agent is not able to select rules when new situations arise, e.g. in uncontrolled environments. Therefore, the SASE agent model is required by *not only* the developmental paradigm, *but also* the conventional non-developmental paradigm, as long as hand-designed rules are not sufficient for the uncontrolled environment (which is typically the case).

Second, the degree of understanding of a rule depends on the degree of detail in the autonomous rule construction. The less the detail, the coarser the understanding. The required granularity of real-time cognition and behavior generation is very fine spatially (e.g. image resolution with attention selection, multi-modality, internal and external sensing) and temporally (e.g. each mental cycle takes 30 ms). For a complex task, the number of steps of task execution and the number of alternatives in each step are both very large. Therefore, the number of possible rules required by the task execution in an uncontrolled environment is astronomical.

If a non-developmental learning scheme is used, a hand-designed task-specific representation (e.g. Markov decision model) is required and then a *model fitter* is required to fit the hand-designed model to data. Specifying the meaning of all the components of the hand-designed representation (e.g. the meaning of all the states) is manually intractable in uncontrolled environments (too many states and a large proportion is unpredictable), let alone the parameters of the representation (e.g. the initial rough estimates of the prior and transition probabilities of all the states, which are required for a learning algorithm to start). Thus, a non-developmental

(task-specific) learning paradigm seems unsuited for the SASE agent, except for some small tasks in a controlled environment.

In contrast, a developmental program is not a *model fitter* for a hand-designed task-specific model, it is a *model generator*. It automatically generates a task-specific model with a large number of internal states (e.g. vector clusters) in controlled or uncontrolled environments. Further, the autonomous mental development paradigm provides an autonomous way to conduct simple to complex shaping of the internal model being autonomously generated. By shaping, we mean that the (human) environments enable the developmental program to generate mostly desirable cognitive behaviors (local models around the experienced events, not too far in the space of all possible events) in a simple to complex manner (similar to a constrained search). Desirable ones for simple cognitive behaviors are developed first before more complex ones (i.e. *scaffolding*). It is also a *self-aware* and *self-effecting* search because near alternatives (slight deviations from the desired ones) have mostly been constructed autonomously and tried by the agent itself to see the effects. This developmental process is not totally random either, because the physical world and human teachers are in the loop (similar to an “*intelligently guided search*”).

In summary, true intelligence, especially the capability to act intelligently in uncontrolled environments, requires a SASE agent. This is true for both developmental and non-developmental paradigms. However, only the developmental paradigm seems suited for realizing complex SASE agents.

3.4. Cognitive development in continuous context

Aristotle (384–322 BC) insisted that the mind is a “blank slate” at birth, a *tabula rasa*, which is, as we know now, not accurate according to studies in developmental psychology.¹⁵ He is right, however, in recognizing that the experiences of an individual are of paramount importance and in identifying the basic principle of association. Descartes’s “rational approach” in the mid-1800s has been discarded by modern scientists, in favor of *observational* or empirical methods of studying the mind. How do we define and measure cognitive capabilities of our robots? Here, we do not adopt a definition of intelligence in terms of “rationality.” Our formulation of cognitive development follows the scientific tradition of careful quantification, *clear definition* and *empirical observation*.

First, cognition requires a discrimination among sensory inputs and a display of the discrimination through actions. The latter is required for the actual use of the cognitive and behavioral capabilities as well as a measurement of such capabilities. Thus, we must address the concept of discriminative capability.

Definition 4. Given a developmental agent at time t_1 , suppose that the agent produces two different action contexts a_1 and a_2 , from two different contexts $C_1 = \{c(t) \mid t_1 \leq t \leq t_2\}$ and $C_2 = \{c(t) \mid t_1 \leq t \leq t_3\}$, respectively. If a_1 and a_2 are considered different by a social group (human or robot), conditioned on C_1 and C_2 , then we say that the agent *discriminates* two contexts C_1 and C_2 in the

society. Otherwise, we say that the agent *does not discriminate* C_1 and C_2 in the society.

The above definition allows for a variation of action context a from the same context C . In other words, even if different robots produce different actions in the same test, they are considered correct if the actions are considered socially equivalent. For example, no two humans have exactly the same voice, but they can pronounce semantically equivalent words. Human categorical perception and equivalence of stimuli, have been extensively studied in psychology (see, e.g. Refs. 18 and 40). A field called psychometrics⁵ has developed systematic scales for measuring cognitive capabilities.

We desire an agent to produce only equivalent actions from all the equivalent contexts. There is a special, but very large, class called the *unknown class* which includes all of the contexts that the agent at this age is not expected to understand. Unlike a traditional classifier, we require a developmental robot to be able to deal with all possible contexts, according to its cognitive maturity. This is in contrast to a traditional robot which deals with only a controlled environment. That is, a developmental agent is supposed to produce a *correct action* even for contexts that it cannot deal with confidently. For example, if the context means “what is this?” the correct action for a baby robot can be “doing nothing” or, for a more mature robot, saying “I do not know” or anything else that is equivalent socially. Of course, human parents may tend to shelter their children from coming into contact with too much of the human adult world, but a developmental program should not assume, exclusively, a pure child’s world during the early development stage.

Definition 5. Given a context domain \mathcal{D} and a set of possible action contexts \mathcal{A} , a *norm* is a mapping N from \mathcal{D} to \mathcal{A} , denoted by

$$N: \mathcal{D} \mapsto \mathcal{A},$$

and it is defined by a social group. The *agent mapping* of an agent at time t is also a mapping, denoted by

$$A(t): \mathcal{D} \mapsto \mathcal{A}. \quad (3)$$

A test for an agent $A(t)$ is to let the agent experience multiple contexts. An evaluation of the performance is a measure that characterizes the agreement of the two mappings N and $A(t)$ through tests.

A mentally developing robot, or developmental robot for short, is an embodied, SASE agent that runs a developmental program following the autonomous developmental paradigm.

Different age groups of developmental robot have corresponding norms. If a developmental robot has reached the norm of a human group of age k , we can say that it has reached equivalent human mental age k .

4. Architecture of the Developing “Brain”

Neisser³³ pointed out that any model of vision that is based on spatial computational parallelism alone is doomed to failure. He proposed a two-stage visual process, which consists of a pre-attentive phase followed by an attentive phase. However, he did not propose a computational architecture for vision. Feldman and Ballard¹⁴ proposed a “100-step rule,” based on the known facts that most neurons compute at a speed of a few milliseconds and that simple visual perceptual phenomena occur in a few hundred milliseconds. Therefore, a biologically plausible algorithm for pre-attentive vision can require no more than 100 steps. A biological vision algorithm is “shallow” and parallel. In the field of computer vision, traditional vision systems are designed for visual perception in a particular environment for a particular visual problem, instead of a general visual capability, although they address the architecture and algorithm levels of detail. John Tsotsos’ study^{44,45} on the complexity of pre-attentive (immediate) vision is a remarkable exception in that it proposed a coarse architecture for a biologically plausible general-purpose vision architecture (for pre-attentive vision). Since the study is meant for complexity analysis, it does not address how his proposed architecture is implemented by an algorithm for general visual environments.

In this section, we will discuss the architecture of the information processor (the “brain” or central nervous system) of SAIL and Dav developmental robots. The key architecture component for AMD is the sensorimotor subsystem. It is a *traditional* view that higher mental activities require an architecture that is very different from a sensorimotor system, but this author is not convinced by this view. There does not seem to be fundamental limitation in the proposed architecture of the sensorimotor system that prevents it from effectively dealing with “higher” mental activities. When co-developed with other sensorimotor systems, an integrated network of sensorimotor systems has the potential to deal with high-level cognitive behaviors such as abstract reasoning and planning. This can be considered a hypothesis for now, since future work is required to demonstrate such a potential. However, our experimental results have shown that simple language skills can be developed from such sensorimotor systems. It is our assertion that complex language skills can be developed from the same sensorimotor architecture (probably with different architecture parameters). The distributed numeric representation in context is essential in scaling up the language (and other) complexity without requiring a different architecture by each different syntax structure.

4.1. Sensory and cognitive mappings

Figure 4 provides a simplified architecture of a multi-level sensorimotor system for development, using visual sensing as an example. This architecture consists of two parallel sensorimotor pathways mediated by subsumption. Each sensorimotor pathway handles a complete mapping from sensory input all the way to motor output. The central pathway (the middle vertical pathway in Fig. 4) from the entire retina,

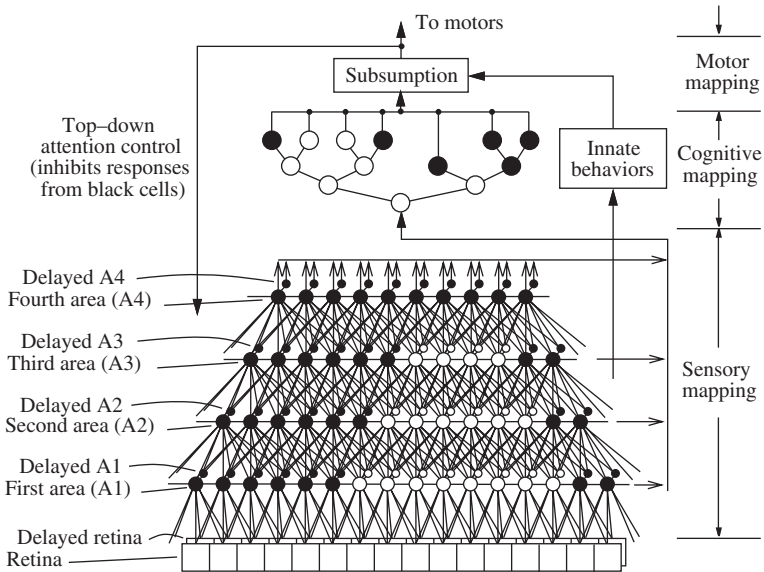


Fig. 4. The flow diagram of a developmental vision system. The responses from black cells are inhibited by the attention control but those from the white cells are passed at this time instance.

through the sensory mapping and the cognitive mapping to the motor mapping (bottom-up) and back to the sensory mapping (top-down) is a major sensorimotor pathway. The innate behavior vertical pathway (shown in Fig. 4 as the “Innate behaviors” block on the right side) is another but simpler sensorimotor pathway. These two pathways are mediated by the subsumption in the motor mapping.

The innate behaviors include simple reflexes (e.g. pain avoidance) and some mechanisms of value-guided exploration (e.g. trials in learning attention selection).^{21,41} A very large proportion of adult cognitive and behavioral capabilities are acquired through, and shaped by, experience. They can also override the innate behaviors, e.g. fighting physical fatigue for a career goal. The subsumption module mediates the action outputs from different sources so that the action from a pathway that is positioned higher (the middle vertical pathway) has a higher priority.⁷

In psychology, there has been a growing literature on the connectionist perspective to the issues of nature-nurture interaction during development (e.g. Ref. 27), the innateness and plasticity (e.g. Ref. 13), as well as more specific issues such as nonlinear developmental trajectories, critical periods, and functional specificity of cortical regions (see, e.g. Ref. 31).

We model the function of a biological cortical region by a (time-varying) mathematical concept called mapping $f_t: \mathcal{X} \mapsto \mathcal{Y}$, where \mathcal{X} is the space of input (nervous inputs) and \mathcal{Y} is the space of output (nervous outputs). Given any vector $x(t) \in \mathcal{X}$, $y(t) = f_t(x(t))$ is a vector in \mathcal{Y} , called the response to $x(t)$. The function f_t itself typically is also updated as a consequence of computing response $y(t)$ from $x(t)$.

A sensory mapping shown in Fig. 4 has two major functions: (i) it provides parallel responses for all possible receptive fields, which are fed into a cognitive mapping for learning, (ii) it executes attention selection control (internal control) signals from the cognitive mapping by suppressing the responses from unattended receptive fields. The attention control is a top-down control from cognitive mapping back to the sensory mapping. The initial receptive field range of each neuron (a unit node in Fig. 4, for feature detection) is hand-designed but is further refined while the connection weights are incrementally computed (learned or developed) from sensing experience, e.g., using the Candid Covariance-free Incremental Principal Component Analysis (CCIPCA).⁶⁰ More detail about sensory mapping is explained in Ref. 65.

A cognitive mapping realizes a mapping $f: \mathcal{X} \mapsto \mathcal{Y}$, where \mathcal{X} is the space of the last contexts and \mathcal{Y} is the space of the primed contexts (predictions, see below). Some major requirements for a cognitive mapping are: (i) it must be constructed incrementally; (ii) have a dynamic number of degree of freedom (parameters) that are automatically determined to fit changing complexities of the mapping exhibited by the experience; (iii) have long term memory to avoid loss of old memory, derive feature subspaces for better generalization from limited training samples; and (iv) have a very low time complexity for each updating when the size of the memory has grown very large.

Incremental Hierarchical Discriminant Regression (IHDR)^{22,54} is used to self-organize the input space of f into a hierarchy of (nested) partitions organized into a tree structure. Although desired output actions may be supplied sometimes, the internal representation of IHDR is not totally supervised and is largely self-organized instead. Each cell in a coarse partition is refined by a fine partition in the next finest level in the tree. At each node, the space is represented by its own automatically developed most-discriminating feature subspace, in which the boundary of cells of the finer partition is determined by the Bayesian estimation. This results in a quasi-optimal generalization boundary, conditioned on the current coarseness of the partition. Such recursive coarse-to-fine partition ends at a node when the number of samples (vector quantized version) it receives is so small that the cluster statistics cannot be estimated reliably. This node is then a leaf node, where a limited number of individual context prototypes (through incremental vector quantization) are kept as context state vectors, each of which is linked to a number of output vector(s) in \mathcal{Y} . The tree structure results in a logarithmic time complexity in the number of leaf nodes in the tree, for each retrieval and update of the tree, making it possible to achieve real-time speed even when the number of context prototypes is very large. It has been systematically demonstrated²² that HDR out-performs many well-known classifiers, including the support vector machines, in a series of high-dimensional tests.

Most traditional task-specific methods have used human-designed, task-specific and environment-specific invariant features, e.g. a particular color for human facial detection. They are called *early invariance* methods (early in sensory processing).

The new developmental architecture introduced here is called *late invariance* in the sense that it fully uses rich information in the sensory data for generality, while the task-specific invariance is achieved through learning, which realizes many (context prototypes) to one (action) mapping using IHDR.

The architecture shown in Fig. 4 is expected to be equally well applicable to other sensory modalities, such as vision,^{55,65} speech^{67,68} and touch,⁶⁴ each with a different set of developmental parameters (e.g., the extent of temporal context, see visual and auditory experiments and citations discussed in later sections). This is practical because representation (e.g. the feature detectors using CCIPCA⁶⁰ and the tree using IHDR^{22,54}) is generated automatically from the sensing experience of that sensing modality.

How does this architecture relate to the major concepts introduced in the previous sections? The coarse architecture (e.g. the partition of sensory, cognitive and motor mappings and their connections) is hand-designed (innate) but the fine architecture (e.g. the connection patterns) and representation (e.g. the weights of connections) are grown and modified incrementally, in real-time, according to the (innate) developmental mechanisms (e.g. CCIPCA and IHDR) and the actual real sensory and motor experience (signals), following the autonomous development paradigm. The architecture is not designed for a particular task (e.g. neither the 3-D position concept nor the 3-D occupancy concept are hand-designed in the internal representation), but for a wide variety of tasks. What tasks the agent ends up learning and executing depends on the actual developmental experience (unknown during the programming time). The introduced architecture is for SASE agent: the top-down attention control is an internal action, acting on the brain itself. The sense of this internal action (internal sensing) has a low degree of freedom (e.g. two for the 2-D retinal position and one for the size of the receptive field) and is sensed by a (virtual) internal sensor not explicitly drawn in Fig. 4. As a rule of thumb, every (internal and external) effector that requires autonomous decisions must have a dedicated internal sensor so that the “brain” can be aware of its status when it attends to it.

4.2. Past and future contexts

A sensorimotor system is a predictor and a doer. At each time instant t , an (AMD) sensorimotor system receives the last context as the input vector:

$$l(t) = (x_l(t), a_l(t)), \quad (4)$$

which contains the *last sensation* $x_l(t)$ and the *last action* $a_l(t)$. A sensorimotor system also needs to predict future sensations and actions. We call them the *primed sensation* x_p and the *primed action* a_p , respectively. The term “prime” is used in psychology to indicate a meaning similar to “predict.” They form what is called the

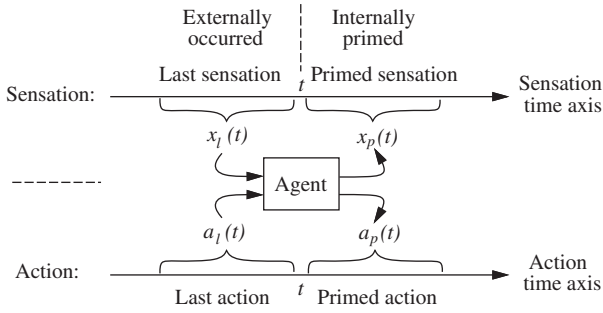


Fig. 5. A developmental agent is a real-time predictor. At any time, the agent has four types of context. It maps the last context $l(t) = (x_l(t), a_l(t))$ to the selected primed context $p(t) = (x_p(t), a_p(t))$ from multiple possible primed contexts.

primed context:

$$p(t) = (x_p(t), a_p(t)). \tag{5}$$

Let \mathcal{P} denote the space of the primed contexts. In more detail, we define four types of context information: last sensation, last action, primed sensation and primed action. They are positioned in the input and output spaces and along the time axis as illustrated in Fig. 5.

Producing a single primed context is not sufficient. This is because the context $l(t)$ is typically not sufficient to predict a unique context. Each context $l(t)$ may correspond to multiple future possibilities $p_1(t), \dots, p_k(t)$ (e.g. left and right turns at a Y junction). This mapping is accomplished by a particular cognitive mapping called reality mapping R :

$$\{p_1(t), \dots, p_k(t)\} = R(l(t)). \tag{6}$$

Thus, the reality mapping R is a mapping from the space of the last context \mathcal{L} to the power set of \mathcal{P} :

$$R: \mathcal{L} \mapsto 2^{\mathcal{P}}. \tag{7}$$

R is developed incrementally through experience. For any $t > 0$ (after birth), it is a total function since it is defined for all elements in \mathcal{L} , but it does not do well for most elements in \mathcal{L} that it has not experienced. It is not an onto function since its range covers only a very small part of $2^{\mathcal{P}}$.

Therefore, we need a value system that selects desirable contexts from multiple primed ones. The value system $V(t)$ takes a set of (e.g. k) contexts from the reality mapping R and selects a single context:

$$V(R(l(t))) = V(\{p_1(t), p_2(t), \dots, p_k(t)\}) = p_i(t), \tag{8}$$

where $1 \leq i \leq k$ and k varies according to experience. In terms of mapping between input and output spaces, the value system is a mapping from the power set of \mathcal{P} to

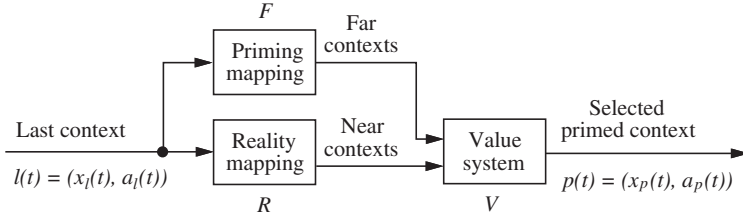


Fig. 6. A recursive view of a simplified sensorimotor system. A more complete architecture is shown in Fig. 7.

the space of \mathcal{P} :

$$V: 2^{\mathcal{P}} \mapsto \mathcal{P}. \quad (9)$$

Further, a single mapping R is not sufficient. We need multiple ones. R maps to near future and F maps to far future as illustrated in Fig. 6. In summary, the three mappings, R , F , and V , accomplish a composite mapping from the space of last contexts \mathcal{L} to the space of primed contexts \mathcal{P} :

$$\mathcal{L} \xrightarrow{R,F} 2^{\mathcal{P}} \xrightarrow{V} \mathcal{P}.$$

Neither of the mappings R and F is static, since both are updated at every time instant t .

4.3. Sensorimotor system as DOSASE MDP

A more detailed block diagram of an example developmental sensorimotor system is shown in Fig. 7.

As shown in Fig. 7, each internal and external action output feeds back, through a delay unit, into the next sensory input. This is required by our SASE agent model: internal and external actions are a target of perception and cognition. The agent must sense and perceive what it does, internally and externally. The input to a sensorimotor subsystem, indicated by the left-most arrow in Fig. 7, is its target for perception and cognition.

A sensory mapping (e.g. SHM⁶⁵) is needed wherever a developmental roboticist finds it necessary to equip the “brain” with an attention selection effector or a dimension reduction processor. As shown in Fig. 7, two sensory mappings are used, the spatial sensory mapping S and the spatiotemporal sensory mapping T . The former is for spatial sensory data attention and the latter takes into account the attention for both space and time. The sensory mapping is developed automatically from the sensed signals.

A developmental system can be represented by a pair $(A(t), D)$, where $A(t)$ is a time varying processor being developed and D is its developer. A nondevelopmental system is represented by a 2-tuple (A_s, B) , where A_s is a static (after

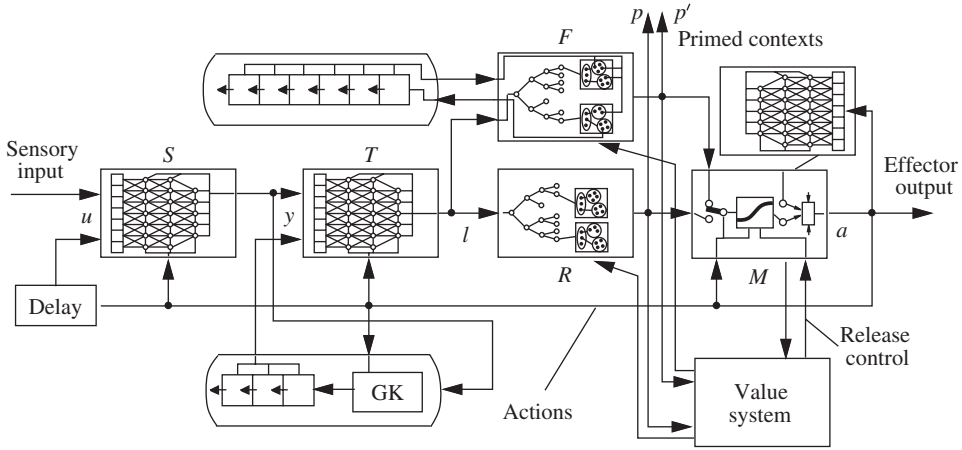


Fig. 7. A block diagram of the architecture of a sensorimotor subsystem. Not all the connections are shown. S is a spatial sensory mapping; T is a spatiotemporal sensory mapping. GK: gate keeper, an internal effector to actively control the update of the last context. R the reality mapping and F the priming mapping, both are implemented by the cognitive mapping engine IHDR. M is the motor mapping. The value system is shown as a block, but it is in fact widely distributed. The motor mapping for high-dimensional stereotyped actions is shown in an attached block.

traditional machine learning) processor and B is its model fitter (B for the Baum–Welch algorithm).

Mathematically, a developing sensorimotor system can be modeled by a developmental Observation-driven SASE Markov Decision Process (DOSASE MDP), defined as follows:

Definition 6. A *Developmental Observation-driven SASE Markov Decision Process* (DOSASE MDP) $A(t)$ is a finite state SASE machine at any time $t = 0, 1, 2, \dots$, and it starts to run at $t = 0$ under the guidance of its developer D . Its observation vector at time t is the last context $l(t)$. The output from $A(t)$ at time t is its selected primed context $p(t) \in \mathcal{P}$ (containing output action). The states l' of $A(t)$ are time-varying vector clusters in a subspace of \mathcal{L} (the space of last contexts). The system $(A(t), D)$ is developmental in the sense that the internal observation-driven SASE MDP is generated and updated autonomously (i.e. developed) through developmental experience; the developer D does not require a given estimate of the *a priori* probability distribution $P(l')$ of \mathcal{L} , nor a given set of states. The states l' dynamically change in meaning and in number. Consequently D does not require a given estimate for the state observation probability $P(l(t) | l'(t))$ nor that for the state observation probability $P(l'(t + 1) | l'(t))$.

We first discuss the similarity. Both $(A(t), D)$ and (A_s, B) use a finite state machine for $A(t)$ and A_s , respectively. For example, the priming mapping F in $A(t)$, implemented by the cognitive mapping engine (e.g. IHDR^{22,54}), maps any

(input) last context $l(t)$ to a list of (far) future contexts:

$$\{p_1(t), \dots, p_k(t)\} = F(l(t)). \quad (10)$$

To do so, it keeps in leaf nodes many time-varying discrete prototypes (vector codes) l' as clusters of the continuous input space \mathcal{L} (based on the sensory experience so far). Given any $l(t) \in \mathcal{L}$, F uses its automatically generated hierarchy of feature subspaces to find the “best matched” prototype l' . l' is similar to a state in the Markov decision process (MDP)²³ in the sense that they both take into account some context. MDP has also been used for a continuous state space (using human-designed features and representation).

The major differences between the DOSASE MDP system $(A(t), D)$ and a traditional MDP system (A_s, B) include: (i) D is an automatic model generator (generates $A(t)$ directly from observation signals) but B is a model fitter (from a given estimated A_s) using, e.g. the *Baum–Welch algorithm*⁴ for which a given good set of initial probability estimates for A_s is necessary to reach an acceptable performance). Although $A(0)$ starts from some innate behaviors, the degree of adaptation of a developing $A(t)$ is much larger than that of a fitted A_s . (ii) $A(t)$ uses a mind representation but A_s uses a world representation (see Section 5). (iii) Each prototype l' in IHDR has an epigenetic representation (defined later) but a state in MDP is symbolic and, thus, cannot be automatically generated without hand-designing the meanings first. (iv) Since the states of $A(t)$ are generated and merged dynamically through time, it can record more flexible context information than the traditional MDP A . Of course, the design of the developer D is considerably more challenging than the model fitter B .

Each prototype l' is associated with a list of primed contexts as output, as indicated by Eq. (10). The *prototype updating queue* of length k keeps the last k visited prototypes so that updating of primed context (not just Q-value) using the Q-learning algorithm⁵⁰ can be done recursively for every prototype in the queue^e from the tail to the head,⁶⁸ instead of only the currently visited prototype in the original Q-learning. This speeds up looking ahead and expands its range. In contrast with the priming mapping F , the reality mapping R does not need such a queue because it only predicts the next-step context only.

The motor mapping M of a sensorimotor system generates concise representations for stereotyped actions (actions repeated many times without much variation). If only a single motor is considered, a motor mapping includes a gating system for each of the single motors as well as the subsumption mechanism for integration from other sensorimotor systems. Through developmental experience, motors that are highly correlated enable the growth of a new part of motor mapping, denoted as an attached block to the basic motor mapping in Fig. 7. The new part of the motor

^eThis is similar to, but not the same as, what is called the eligibility trace,⁴¹ due to value-guided exploration, human interactions, primed sensations that are not in the traditional reinforcement learning framework.

mapping plays the corresponding role of the gating system, but it is for correlated multi-motor actions.

In supervised learning, the value (motivational) system is not needed (unique given action output). But the robot is also allowed to perform autonomous learning. The innate value system²¹ uses rewards and novelty through Q-learning to approximate the value for each context so that a learned value system is developed. The later learned value system is further based on an understanding of social norms, e.g. what the parents are happy about or what is right and what is wrong,³⁴ through further development without a need for reprogramming (which needs further experimental studies of course).

5. Internal Representation

The term “internal representation” refers to the representation used internally (the central nervous system or the “brain”) by the agent.

5.1. *World and mind concepts*

In the current AI literature, the distinction between world concepts and mind concepts have been largely ignored. One main reason is that it is the human programmer who designs a representation (e.g. Soar,²⁵ ACT-R,² Markov Decision Process (MDP)²³) and, therefore, it is assumed that the designed representation is correct for the modeled part of the world. For this reason, symbols are commonly used for representation. This type of world representation is effective for dealing with a contained fully-modeled problem. However, it limits the system’s capability to go beyond those that the fixed set of symbols can represent.

We need to distinguish between the actual physical world and the mental effects that it causes. In general, they are not the same.

Definition 7. 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. A *mind concept*^f of an agent is an internal representation, internal with respect to the nervous system (including the brain) of the agent, as a compounding effect of the developmental program and the agent’s experience.

Figure 8 illustrates world and mind concepts. A world concept is about the world, no matter whether the agent understands it or not, or it is true or not. A mind concept typically corresponds to a partial observation of objects in the world. For example, “in front of the agent there is an apple” is a world concept about the current world. It is about a fact of the world, no matter whether we call

^fThe term “mind” is used for the ease of understanding without unnecessarily coining new words. We do not claim that it is the same as the human mind.

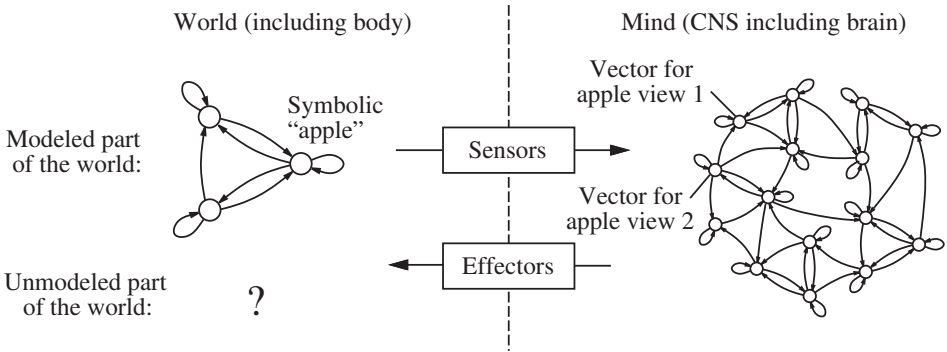


Fig. 8. World and mind concepts. On the left, a part of the world is modeled by a human-designed representation, which represents world concepts. On the right, mind-centered numeric representation is automatically generated which represents mind concepts.

the object an apple or something else. Suppose it is true that there is an apple. If a robot sensed the apple and determined that “in front of me there is a pear,” then “in front of me there is a pear” is a mind concept.

Definition 8. A *world-centered representation* is such that every item in the representation corresponds to a world concept. A *mind-centered representation* of an agent is such that every item in the representation corresponds to a mind concept of the agent.

There is no one-to-one correspondence between a world-centered representation and a mind-centered representation. Typically, many mind-centered representations correspond to the same world-centered representation. For example, many views of the same human face result in many mental prototypes in the brain.

A mind-centered representation is specific to a particular agent (mind). It can only represent mind concepts well. A mind concept is related to phenomena observable from the real world, but it reflects the reality only partially because of the limited sensing capability. It does not necessarily reflect reality correctly either. It can be an illusion or totally false.

5.2. Symbolic and numeric representations

A world concept can conveniently use a symbolic representation for understanding by humans. This is because symbols are created by humans to communicate among humans.

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

For example, Apple = (shape, weight, color) is a symbolic representation of a class of objects called apple. Apple-1 = (round, 170 g, red) is a symbolic representation of a concrete object called Apple-1.

A typical world-centered symbolic representation has the following characteristics:

- (i) each component in the representation has a predefined meaning about the object in the external world;
- (ii) each attribute is represented by a unique variable in the representation;
- (iii) the representation is unique for a single corresponding physical object in the external environment.

These characteristics have been a major reason for the representation to be used widely in knowledge representation, databases, expert systems, and many other traditional AI systems. In a non-developmental approach, it is convenient for a mind-centered concept to use a symbolic representation. However, for developmental robots, it is not possible to use symbolic representation for mind-centered concepts, as we will explain in the following section.

A mind-centered numeric representation is not necessarily about any particular object in the environment. It is mind-centered, grown from the body's sensors and effectors. The early sensory form of such a representation is called *iconic representation*¹⁹ and the later form *categorical representation*. Harnad,¹⁹ Brooks⁸ and others have pointed out the importance of grounding. For conciseness, we propose calling mind-centered numeric representation *epigenetic*⁸ representation. An epigenetic representation is formed from sensory and effector signals (thus, the 'epi' part) and the developmental program (thus, the genetic part), which enables the formation of feature representation according to the statistics of input signals.

Definition 9. An *epigenetic representation* is defined recursively as a vector form $v = (v_1, v_2, \dots, v_n)$, where v (optional) denotes the vector (e.g. neuron) 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 these two types and other (intermediate) epigenetic representations.

The *world-centered* and *mind-centered* representations are the same only in some trivial cases, e.g. where the entire external world is the only single object for cognition. On the other hand, an effector-centered representation (the vector of motor control signals) can correspond to a world object well in some cases, 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 (mind-centered) representation. However, since the generated actions are not

⁸A term used often in developmental biology and developmental psychology.

unique, given different sensory inputs of the same object, it is difficult and unnecessary for the brain (human or robot) to arrive at a unique representation from a wide variety of sensory contexts that correspond to the same single object. According to the above discussion, it seems unlikely for a developmental being (human or robot) to develop a monolithic (mind-centered but having a one-to-one correspondence with a world object) internal representation.

Therefore, a symbolic representation is not suited for a developmental program, but a high dimensional mind-centered numeric representation (i.e. epigenetic representation) is and such a representation should be everywhere in the developing “brain.”

6. Experiments

6.1. *Developmental system projects*

Our decade-long effort in enabling machines to grow their perceptual, cognitive, and behavioral capabilities has gone through four systems: Cresceptron (1991–1995), SHOSLIF (1993–2000), SAIL (1996–present) and Dav (1999–present).

Cresceptron is an interactive software system for visual recognition and segmentation.⁵² 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 vision systems to be trained to recognize and segment complex objects of very different types from natural, complex backgrounds. Although Cresceptron is a general developmental system, its efficiency is problematic.

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 the Principal Component Analysis (PCA) followed by the Linear Discriminant Analysis (LDA), for better generalization. It uses 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 Engineering Building (covering 136×116 square meters) in real-time using only video cameras, without using any range sensors.⁵³ All the real-time computing (refreshing rate 6 Hz) was performed by a Sun SPARC-1 (33 MHz) Workstation. Therefore, SHOSLIF is very efficient for real-time operation. However, it is not an incremental learning method.

The **SAIL** robot, shown in Fig. 9, is the next generation platform after SHOSLIF. The objective of this project is to accomplish real-time incremental development for robot perceptual and behavioral capabilities.^{55,56} It is a wheel-driven untethered mobile robot with a single robot arm. It has a total of 13 DOF. Its sensors include two color video cameras (each can pan and tilt individually), microphones, a laser range scanner (not used for the navigation experiments discussed here),

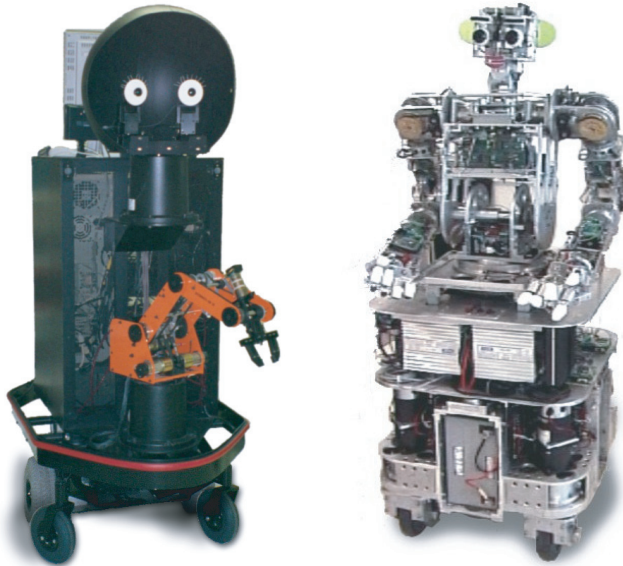


Fig. 9. The SAIL robot (left) and Dav robot (right).

and an array of touch sensors and micro-switches. Its computational resources are all onboard, including dual Pentium IV 2.1 GHz, 1 Gb RAM memory, 50 G SCSI hard disk drives, and an array of device drivers. It weighs 202 Kg. Most of the experiments reported here were conducted on the SAIL robot.

The **Dav** robot (Fig. 9) is an anthropomorphic robot, constructed in-house at Michigan State University as a next-generation test-bed for experimental investigations into autonomous mental development.^{17,63} This general-purpose humanoid platform consists of a total of 43 degrees of freedom (DOF), including the wheel-driven 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 sensors, including visual (two color video cameras), auditory (microphones), a laser range scanner, haptic sensors, and somatic sensors (e.g. strain gauges). It is untethered and mobile with all the computational resources onboard, including quadruple Pentium III Xeon 700 MHz CPU, 2 Gb RAM memory, 100 Gb SCSI drives, 11 embedded Motorola PowerPCs 555 40 MHz processors, CAN bus for communication among CPUs and embedded processors, wireless networks, and a 440 Amp-Hour 12 V battery power supply. It weighs 242 Kg.

Three types of learning modes have been implemented on SAIL with the SAIL-3 developmental program: learning by supervised learning, reinforcement learning, and communicative learning. In the following sections, we report some experimental results. All the learning experiments presented here were conducted incrementally (about 30–100 ms per cycle) online in real-time, except those stated otherwise. The

“brain” of the robot is totally signal driven (from sensors and effectors), generated by the developmental program as a model generator. There is no need for an initial guess (e.g. weights of connections). In the experiments described here, supervised learning was used to generate desired behaviors and, thus, the innate behavior module was not used (it was used to generate the Boltzmann exploration⁴¹ in other studies, e.g. development of SAIL’s value system²¹ and SAIL’s reinforcement speech learning⁵⁹). Movies are available at www.cse.msu.edu/~weng/research/LM.html.

6.2. Developmental recognition from occluded views

We have designed and implemented a sensory mapping, called “Staggered Hierarchical Mapping (SHM)” shown in Fig. 4, and its developmental algorithm.⁶⁵ Suppose that a face is occluded and, thus, only attention to the unoccluded partial view enables successful recognition based on the partial view, provided that the partial view can uniquely determine the identity of the face. This implies that the agent must actively select attention to the unoccluded part during the learning session. This is the goal of the experiment illustrated in Fig. 10.

The goal of the experiment, not conducted in real-time, is to study the effectiveness of sensory mapping (SHM) for attention selection under the control of attention signals generated from cognitive mapping, as shown in Fig. 10. The experiment was organized as follows. In the training session, a series of unoccluded face images is presented to the system with class labels (name of the person). The system

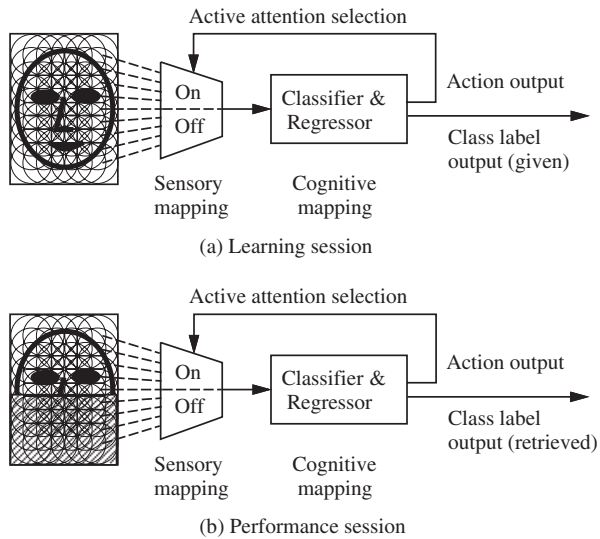


Fig. 10. Active attention during learning and performance sessions enables recognition of occluded faces.

takes upper (U) and lower (L) views, controlled by the supervised attention control (to focus on the module to be tested). In other words, the attention control also conducts supervised learning in this setting. The cognitive mapping learns two action outputs from the currently sensed image, (i) the required attention selection (upper or lower view), and (ii) the class label of the face image. In the performance session, the learned attention control behavior controls the attention selection via SHM sensory mapping, which feeds the response to the following IHDR cognitive mapping. If only the upper view is available (not occluded), the result is called U. Similarly, if only the lower view is available (not occluded), the result is called L. If the system feeds the upper view and the lower view as an integrated long response vector into the IHDR classifier (upper view and lower view is occluded individually at two consecutive views), the result is called U + L.

If a system is passive (without active attention selection), it learns the global view (not occluded) but in the performance session it tries to match the input (occluded) U or L view with the learned global-view (not occluded) prototype. This is called monolithic vision. If we use the nearest neighbor method (NN) for the monolithic vision case to find the prototype, the result is called monolithic + NN.

The experiment used a face set from the Weizmann Institute in Israel. The set was taken from 28 human subjects, each having 30 images with all possible combinations of two different expressions, three lighting conditions and five different facial orientations. The results are summarized in Table 1. They clearly demonstrated the necessity of active attention using the sensory mapping, whose recognition rate (SHM + HDR for U and L cases) is significantly higher than the case without sensory mapping (Monolithic + NN for U and L cases) in the presence of occlusion.

In this experiment, the programmer did not know the task during the programming time, i.e. he did not know that the robot would recognize human faces (or something else), nor did he know that the objects to be recognized would be 2D patterns or 3D objects. The sensory mapping SHM, as shown in Fig. 4, was developed from viewing over 5,000 natural images.⁶⁵ Therefore, the internal representation of SHM is of general purpose: it represents the structure of natural scenes (via camera) using the statistical distribution of image inputs (by CCIPCA), but not other images that it has not observed. When the sensory mapping is mature, the system starts to develop the cognitive mapping from the output of the sensory mapping to the action output (attention control and class label). Both actions are learned through supervised learning. Attention control is an internal action (acting on the “brain”), which normally does not allow supervised learning (the internal effector

Table 1. Summary of recognition under occlusion.

Method	Recognition rate		
	U (%)	L (%)	U + L (%)
Monolithic + NN	51.43	75.83	82.38
SHM + HDR	92.86	95.95	98.57

is not accessible from the environment). Since reinforcement learning takes a significant amount of training time, we used supervised learning to speed up the learning in this study.

SHM + HDR for the U + L case indicates a “programmed way” of integrating multiple views through time, instead of an autonomous way. Autonomous integration of discrete input frames is studied in Sec. 6.6.

6.3. *Developmental vision-guided navigation*

In the experiment of vision-guided navigation,⁵⁵ a human teacher taught the SAIL robot by taking it for a walk along the corridors of MSU’s Engineering Building. Force sensors on the robot body sense the push action of the teacher and its two drive wheels comply by moving at a speed that is proportional to the force that is sensed on each side. In other words, the robot performed supervised learning in real-time.

The IHDR mapping algorithm processes the input image in real-time. It derives features that are related to the action but disregards features that are not. The human teacher does not need to define features. The system runs at about 10 Hz, ten updates of navigation decisions per second. In other words, for each 100 milliseconds, 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 four trips along slightly different trajectories along the corridors, 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 corridor, without the need for “hand-leading.” We found that about ten trips were sufficient for the SAIL robot to navigate along the corridors, using only vision, without using any range sensors. Figure 11 shows some images that the robot saw during the navigation.

Here, the developmental program does not contain any information about what kind of scenes that the robot will sense or what behaviors will be needed. The

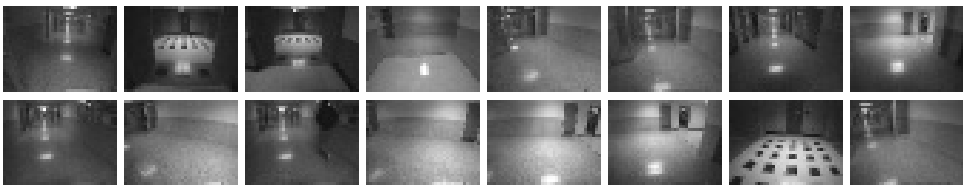


Fig. 11. A subset of images sensed by the SAIL robot in the autonomous navigation, showing the wide variation of the scenes that the robot needs to learn.

program generates a hierarchy of the most discriminating feature subspaces in the IHDR tree. Therefore, the fine architecture (e.g. the interconnection among nodes) and the representation (e.g. the discriminating feature basis in all the non-leaf nodes) are functions of the input signals. What is “innate” is the hand-designed scheme of developing the IHDR tree using input and output signals, but the IHDR tree actually developed is neither totally “innate” (without a need for experience) nor totally “learned” (without any innate component). The developmental program is a model generator, generating the information processor (i.e. IHDR tree as a model of the environment and the navigation task) “on the fly,” incrementally, in real-time, without the need for human intervention into the internal representation during all training and performance sessions (running the same program in a single developmental mode).

Since the processor can be generated automatically “on the fly,” the SAIL robot was moved outdoors (around the Engineering Building) for autonomous navigation, without the need for reprogramming, and performed with limited success.⁶⁶ A major difference between indoor and outdoor environments is the degree of lighting variation. We trained the SAIL robot outdoors during different times of day (10 am, noon, 2 pm, 4 pm, 6 pm, etc.) and under different types of weather conditions (e.g. sunny and overcast), so that the robot became used to a wide variety of lighting variations (which caused hard-to-predict effects such as shadow casts from trees).

6.4. *Developmental speech learning*

Our developmental speech learning is very different from traditional speech learning^{38,48} in the following sense: (i) the continuous auditory streams have not been segmented and labeled (thus, autonomous learning is possible); (ii) during learning, the entire auditory system must listen to everything (for autonomous learning), in contrast to traditional supervised learning where each designed model (e.g. for a word “good”) listens to only segmented speech corpora of the *single* class that it is designed to recognize (e.g. various utterances of the word “good”). For example, if a traditional HMM model for recognizing “good” was allowed to listen to many other words during training, it cannot tell “good” from other sounds; (iii) no syntax is involved during programming (e.g. the system can learn words and phrases from multiple languages concurrently).

The above points (i) and (ii) are necessary for autonomous speech learning. No traditional speech recognition methods can deal with them. Point (iii) is necessary for the task-nonspecificity nature of development. Semantics and syntax are associated with the real-world grounded experience. In other words, the robot performs grounded, autonomous language acquisition (words and phrases only so far), which was impossible before with traditional approaches.

Similar to learning vision-guided navigation, the SAIL robot can learn to follow voice commands through physical interaction with a human trainer.⁶⁷ In the early supervised learning stage, a trainer spoke a command (a word or a continuous

Table 2. Performance of the SAIL robot in developmental speech learning.

Commands	Go left	Go right	Forward	Backward	Freeze
Correct rate (%)	97.1	91.3	93.8	100.0	80.0
No. of tests	35	23	65	7	5
Commands	Arm left	Arm right	Arm up	Arm down	Hand open
Correct rate (%)	100.0	90.0	100.0	100.0	90.0
No. of tests	10	10	10	10	10
Commands	Hand close	See left	See right	See up	See down
Correct rate (%)	90.0	100.0	100.0	100.0	100.0
No. of tests	10	10	10	10	10

phrase) 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. In later stages, when the robot could explore more or less on its own, the human teacher used reinforcement learning by pressing its “good” or “bad” button to encourage and discourage certain actions. Typically, after about 15 to 30 minutes of interaction with a particular human trainer, the SAIL robot could follow commands with a success rate of about 90%. Table 2 shows the voice commands learned by the SAIL robot and its online test performance.

A developmental robot should not be expected to recognize sophisticated, long, continuously spoken sentences in an early developmental stage, neither should a human baby. Section 6.6 explains why “arranged experience” (e.g. separately spoken commands) is important for scaffolding. The major breakthrough here is *autonomous auditory learning* characterized by points (i), (ii) and (iii) above, which conjunctively make *autonomous scaffolding* possible.

6.5. *Developmental communicative learning*

With supervised learning, the human teacher must provide actions in real-time. With reinforcement learning, it takes a significant amount of time for the robot to generate a desired action. With the communicative learning, the human teacher can directly state:

- (i) a desired action in the current context (our experiment);
- (ii) whether the current action is good (our experiment);
- (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).

In this section, we will describe (i) and (ii). The next section will describe (iii). Realizing effective teaching for material type (iii) using sophisticated human language



Fig. 12. The SAIL robot navigated autonomously using its vision-based sensorimotor skills that were acquired through online real-time developmental learning. It perceived the scene from its video cameras without using any range sensors.

and type (iv) via the communicative learning mode is exciting future research direction of AMD.

Recently, we successfully implemented the new communicative learning mode on the SAIL robot for teaching material types (i) and (ii) through autonomous development. First, in the grounded language acquisition stage, we taught the SAIL robot simple verbal commands (phrases), such as “go forward,” “turn left,” “turn right,” “stop,” “look ahead,” “look left,” “look right,” etc. and an evaluation of the current action “good,” “bad”, 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 and encouraged or discouraged the robot’s autonomous action by stating “good” or “bad.” 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.” That way, we did not need to physically touch the robot during training and instead used much more sophisticated verbal commands. This made training more efficient and more precise. Figure 12 shows the SAIL robot navigating in real-time along the corridors of the Engineering Building, at a typical human walking speed. The next section describes a more sophisticated example of communicative learning.

6.6. *Scaffolding: Transfer and chaining*

We first define scaffolding:

Definition 10. *Scaffolding* is the process of using developed simple capabilities to further develop more complex capabilities, through further experience (with or without a teacher), without the need of manual modification of the developmental program.

Human teachers typically “arrange experience,” rather than didactic teaching. Lev Vygotsky⁴⁷ proposed the concept of “zone of proximal development” (PZD), which is a latent learning gap between what a child can do on his or her own and

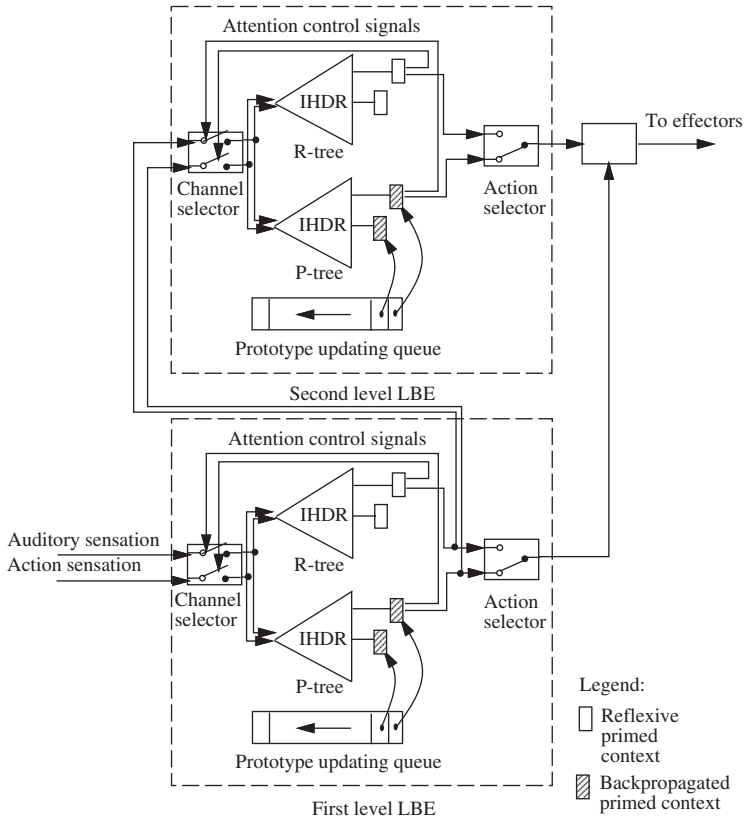


Fig. 13. Integrated architecture with two sensorimotor systems, one lower (less abstract) and the other higher (more abstract) to accomplish developmental learning for more complex skills such as transfer and chaining.

what can be done with the help of a teacher. Wood, Burner & Ross⁶² used the term “scaffolding” to describe such an instructional support through which child can extend or construct current skills to higher levels of competence. Through this process, the scaffolding (arranged experience) is slowly removed.

A powerful developmental program should have mechanisms for scaffolding embedded since a collection of flat (non-hierarchical) structure sensorimotor modules cannot enable complex perceptual, cognitive and behavioral capabilities.

We have designed and implemented a hierarchical developmental learning architecture (Fig. 13), which enables a robot to develop complex skills after acquisition of simple ones.⁶⁸ The major architecture mechanism that makes this possible includes priming and attention, one that realizes *chained secondary conditioning*. However, the mechanism described here is more complex, belonging to what is called *transfer*,¹¹ transferring multiple cognitive and behavioral skills learned in a setting to new settings and chaining them by taking into account new contexts.

A transfer-and-chaining process can be written mathematically as:

$$C_c \rightarrow C_{s1} \rightarrow A_{s1} \rightarrow C_{s2} \rightarrow A_{s2} \Rightarrow C_c \rightarrow A_{s1} \rightarrow A_{s2}, \quad (11)$$

where C_c is the composite (verbal) command, C_{s1} and C_{s2} are (verbal) 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. 14), the SAIL robot learned to combine individually instructed movements into a composite one invoked by a single verbal command without any reprogramming (Fig. 15). To solve the problem of missing context in transfer and chaining, we modeled a primed context as the follow-up sensation and action of a real context. By back-propagating 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.

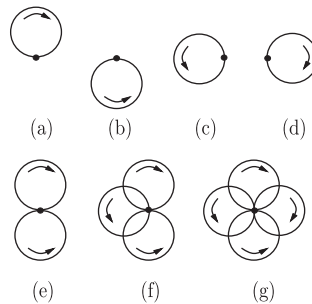


Fig. 14. The gripper tip trajectories of the SAIL robot. (a)–(d) are basic actions, each of which starts from the black dot. (e)–(g) are composite actions by transferring and chaining some or all of the basic ones.



Fig. 15. The SAIL robot learned longer and more complex composite skills through transfer-and-chaining based on previously learned simpler and shorter skills, while it was interacting with human trainers in real-time.

Again, we can see how task-nonspecificity is realized for scaffolding. The programmer does not need to know what complex tasks the robot will execute, let alone to understand them. As long as the teacher (or environment) provides temporally (somewhat) consistent occurrences of sensory events (not pre-defined symbolic ones), the developmental program is able to establish the corresponding association in the developing “brain” and to prime the corresponding contexts in the future under a similar context, allowing the scaffolding of more and more complex cognitive and behavioral capabilities.

The SAIL scaffolding mechanism enables the association to take place in longer temporal scales and coarser spatial resolution scales, which facilitates abstraction. Sharing of simpler skills (a form of autonomous chunking) takes place automatically, within a task and across tasks, when developing more complex skills.

6.7. *Dav: Range-based collision avoidance*

The Dav robot was used to test collision avoidance using its Sick laser range scanner. In each time frame, the Sick scanner produces 360 laser rays, spreading evenly over a horizontal plane (0.5° resolution). Each number in the frame (vector) represents the range (distance) from the Sick scanner to the obstacle that intercepts the corresponding laser ray. The Sick laser scanner is mounted on the Dav robot with a slight tilt downwards, so that the laser plane can detect low obstacles when Dav moves forward.⁶³

In this experiment,⁶⁴ IHDR was used to learn the mapping from the input range vector r to the desired heading direction θ and speed v , which are supplied online interactively by a human teacher via a graphic user interface. To reduce the amount of interactive training needed, an attention mechanism is used to suppress some parts of the range vector when it is necessary, before the result is fed into IHDR for learning. The attention selection controller (regarded as innate reflex) was programmed to behave this way: if all the readings are larger than a threshold T , all the readings are passed because no special attention is needed when all obstacles are far away. If there are some range readings that are less than T , they correspond to nearby objects and are passed without modification, but all the other readings (corresponding to faraway obstacles) are replaced by the mean range value. This way, nearby objects are attended to and faraway ones are not, unless there is no nearby object.

First, to reach a quantitative evaluation with ground truth, we recorded 1,917 training samples each consisting of an input-output pair. We performed ten-fold leave-one-out tests. The average error rates over ten tests are shown in Table 3. Without attention selection, the heading error increased to 0.11 with standard deviation 0.30 and the speed error increased to 0.0079 with standard deviation 0.073. Comparing the results, we can see that both the mean and deviation of error were reduced by introducing the attentional mechanism. Our continuous tests showed that the version without attention selection ran into a complex array of objects but the one with attention selection did not.

Table 3. The simulation results of range-based obstacle avoidance with attention selection.

Parameter	Range	Mean of error	Deviation of error
Heading θ	$[0, \pi]$	0.094	0.27
Speed v	$[0, 1.0]$	0.0071	0.074

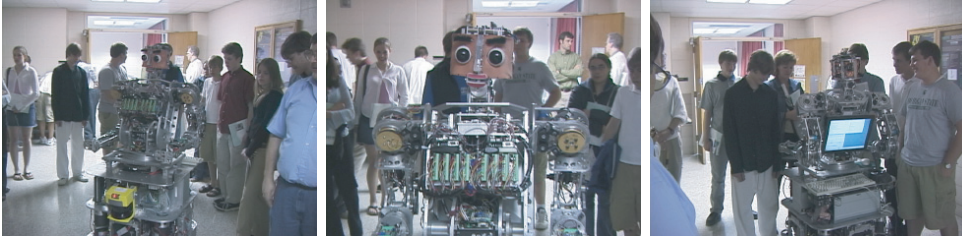


Fig. 16. Dav moved autonomously in a corridor crowded with people, using its laser range scanner.

The Dav robot has been repeatedly tested for learned range-based collision avoidance behavior and the performance has been very satisfactory. For example, during a visit by high school students, as shown in Fig. 16, Dav reliably navigated in this dynamic changing environment without hitting obstacles, static ones or moving people. It is worth noting that most of the testing scenarios were not the same as the training scenarios.

The appropriate sensors can make some tasks easier to learn. For collision avoidance, learning using a range sensor is easier than, e.g. using a pair of stereo cameras, at least for appearance-based methods (mapping directly from normalized input image). Of course, the learning is still not trivial, due to the wide variety of range maps. On the other hand, a pair of stereo images provide additional information that is very useful for other tasks, such as recognition.

Due to space constraints, some major recent experiments cannot be described here, e.g. SAIL's object permanence experiment,⁵⁸ auditory and visual integrated learning for rotating objects⁶⁹ and development of the motivational system.²¹

7. Other Related Work

A type of general models put emphasis on sequential decision making, where the context is represented by a symbolic state. Soar²⁵ and ACT-R² were motivated by state-based interactive cognitive models. The Markov Decision Process (MDP)²³ and MDP-based reinforcement learning⁴¹ are state-based statistical learning models.

Another type of general models put emphasis on perception, modeled as high-dimensional regression. The neural network based methods, such as ALVINN³⁵ and ROBIN,³⁶ can, in principle, be applied to indoor environments. However, the local

minima and loss-of-memory (interference) problems with artificial feedforward neural networks and the local minima problem with the radial basis function make them problematic in low-contrast, fine-detail indoor scenes (see Ref. 10 for a detailed comparison). State-based SHOSLIF¹⁰ provided a general model with both emphases, sequential decision making and perception.

The third type of studies modeled some aspects of cognitive development. BAIRN⁴⁹ (a Scottish word for “child”) is a symbolic self-modifying information processing system as an implementation for a theory of cognitive development. Drescher¹² utilized the schema, a symbolic tripartite structure in the form of “context-action-result.” Some behavior-based robots, such as Cog⁹ and Kismet⁶ at MIT, performed interesting real-time social interactions with humans (some components of Cog were offline learned). David Touretzky’s Skinnerbot⁴³ performed action chaining successfully, through pre-programmed symbolic representation. The Darwin V robot¹ modeled the development of more complex vision-invoked behaviors from simpler pleasure seeking and pain avoidance behaviors. Since its goal was to verify the inter-cortical association from the experience of a real-world device (in a controlled block environment), Darwin V did not address the practical issues of generating new representations for complex uncontrolled human environments. A few more recent studies simulated infant exploratory behaviors with learning, such as using programmed reflexes to explore,^{28–30} using preferred grasping patterns,⁶¹ using changes in retinal resolution and environmental complexity,³² and using histogram association in audio and visual signals.³⁷ Levinson and his co-workers recently demonstrated interactive learning of verbal commands by a mobile robot.²⁶

8. Conclusions

This paper introduces a theory and presents experimental results for a new kind of robot — developmental robots that can develop their cognitive and behavioral skills, autonomously, incrementally, online, through real-time interactions with the environment without pre-designed task-specific representation. The SASE agent model is useful for both nondevelopmental and developmental agents, but it seems that only developmental agents are able to develop the SASE model effectively. The theory reasons that although a world-centered symbolic representation is still useful for simulating some aspects of development, it is not suitable for autonomous mental development, for which a mind-centered numeric representation is suitable. These concepts may also have implications to biological brains, since the brain is also a developmental entity.

The architecture of intelligent agents is an important yet very challenging subject. The architecture outlined here seems to be the first general, task-nonspecific, developmental architecture that generates task-specific internal fine architecture (e.g. the tree structure in IHDR) and representation (e.g. weights in SHM and IHDR) online and yet is suited for an open number of simple-to-complex tasks.

Some experimental results of the proposed theoretical framework have been tested on the SAIL and Dav robots for multiple tasks. The internal representation of the systems is automatically generated based on the co-working of (innate) developmental mechanisms and the (learned) experience. It appears that we have reached a theoretical and practical starting point of a promising new direction of developmental robotics. While there are still plenty of practical and theoretical questions awaiting investigation, this work opens up a wide range of opportunities for future exciting research and applications.

Acknowledgments

This 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, Y. Chen, D. Cherba, C. Evans, J. D. Han, W. S. Hwang, X. Huang, K. Y. Tham, S. Q. Zeng, N. Zhang and Y. Zhang for their contributions to the SAIL and Dav projects as cited.

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, *Cerebral Cortex* **8**(4), 346–361 (1998).
2. J. R. Anderson, *Rules of the Mind* (Lawrence Erlbaum, Mahwah, New Jersey, 1993).
3. R. C. Arkin, *Behavior-Based Robotics* (The MIT Press, Cambridge, Massachusetts, 1998).
4. L. E. Baum, An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, *Inequalities* **3**, 1–8 (1972).
5. N. Bayley, *Bayley Scales of Infant Development*, 2nd edn. (Psychological Corp., San Antonio, TX, 1993).
6. C. Breazeal and B. Scassellati, Infant-like social interactions between a robot and a human caretaker, *Adaptive Behavior* **8**, 49–74 (2000).
7. R. A. Brooks, A robust layered control system for a mobile robot, *IEEE J. Robotics and Automation* **2**(1), 14–23 (1986).
8. R. A. Brooks, *Cambrian Intelligence: The Early History of the New AI* (MIT Press, Cambridge, Massachusetts, 1999).
9. R. A. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati and M. M. William, The Cog project: Building a humanoid robot, in *Computation for Metaphors, Analogy and Agents*, ed. C. L. Nehaniv, Springer Lecture Notes in Artificial Intelligence, Vol. 1562 (Springer-Verlag, New York, 1999).
10. S. Chen and J. Weng, State-based SHOSLIF for indoor visual navigation, *IEEE Trans. Neural Networks* **11**(6), 1300–1314 (2000).
11. M. Domjan, *The Principles of Learning and Behavior*, 4th edn. (Brooks/Cole, Belmont, California, 1998).
12. G. L. Drescher, *Made-Up Minds* (MIT Press, Cambridge Massachusetts, 1991).

13. J. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi and K. Plunket, *Rethinking Innateness: A Connectionist Perspective on Development* (MIT Press, Cambridge, Massachusetts, 1997).
14. J. A. Feldman and D. H. Ballard, Connectionist models and their properties, *Cogn. Sci.* **6**(3), 205–254 (1982).
15. J. H. Flavell, P. H. Miller and S. A. Miller, *Cognitive Development*, 3rd edn. (Prentice Hall, New Jersey, 1993).
16. S. Franklin and A. Graesser, Is it an agent, or just a program?: A taxonomy for autonomous agents, in *Intelligent Agents III, Lecture Notes on Artificial Intelligence*. (Springer-Verlag, Berlin, 1997), pp. 21–35.
17. J. Han, S. Zeng, K. Tham, M. Badgero and J. Weng, Dav: A humanoid robot platform for autonomous mental development, in *Proc. IEEE 2nd Int. Conf. Development and Learning (ICDL 2002)* (MIT Press, Cambridge, Massachusetts, June 12–15, 2002), pp. 73–81.
18. S. Harnad, *Categorical Perception: The Groundwork of Cognition* (Cambridge University Press, New York, 1987).
19. S. Harnad, The symbol grounding problem, *Physica D* **42**, 335–346 (1990).
20. H. Hexmoor, L. Meeden and R. R. Murphy, Is robot learning a new subfield? The RoboLearn-96 workshop, *AI Magazine* (Winter 1997), pp. 149–152.
21. X. Huang and J. Weng, Novelty and reinforcement learning in the value system of developmental robots, in *Proc. 2nd Int. Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems (EPIROB'02)*, Edinburgh, Scotland, August 10–11, 2002, pp. 47–55.
22. W. S. Hwang and J. Weng, Hierarchical discriminant regression, *IEEE Trans. Pattern Analysis and Machine Intelligence* **22**(11), 1277–1293 (2000).
23. L. P. Kaelbling, M. L. Littman and A. W. Moore, Reinforcement learning: A survey, *J. Artif. Intell. Res.* **4**, 237–285 (1996).
24. E. R. Kandel, J. H. Schwartz and T. M. Jessell (eds.), *Principles of Neural Science*, 4th edn. (McGraw-Hill, New York, 2000).
25. J. E. Laird, A. Newell and P. S. Rosenbloom, Soar: An architecture for general intelligence, *Artif. Intell.* **33**, 1–64 (1987).
26. Q. Liu, S. Levinson, Y. Wu and T. Huang, Robot speech learning via entropy guided LVQ and memory association, in *Proc. INNS-IEEE Int. Joint Conf. Neural Networks*, Washington, DC, July 14–19, 2001, pp. 2176–2181.
27. J. L. McClelland, The interaction of nature and nurture in development: A parallel distributed processing perspective, in *International Perspectives on Psychological Science*, Vol. 1: Leading Themes, eds. P. Bertelson, P. Eelen and G. d'Ydewalle (Erlbaum, Hillsdale, NJ, 1994), pp. 57–88.
28. G. Metta and P. Fitzpatrick, Better vision through manipulation, in *Proc. 2nd Int. Workshop on Epigenetic Robotics*, Edinburgh, Scotland, August 10–11, 2002, pp. 97–104.
29. G. Metta, G. Sandini and J. Konczak, A developmental approach to sensori-motor coordination in artificial systems, in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Vol. 4, October 11–14, 1998, pp. 3388–3393.
30. G. Metta, G. Sandini and J. Konczak, A developmental approach to visually-guided reaching in artificial systems, *Neural Networks* **12**(10), 1413–1427 (1999).
31. Y. Munakata and J. L. McClelland, Connectionist models of development, *Development Science* **6**(4), 413–429 (2003).
32. Y. Nagai, M. Asada and K. Hosoda, A developmental approach accelerates learning of joint attention, in *Proc. IEEE 2nd Int. Conf. on Development and Learning (ICDL 2002)* (MIT Press, Cambridge, Massachusetts, June 12–15, 2002), pp. 277–282.

33. U. Neisser, *Cognitive Psychology* (Appleton-Century-Crofts, New York, 1967).
34. J. Piaget, *The Moral Judgement of the Child* (Simon & Schuster, New York, 1997).
35. D. A. Pomerleau, ALVINN: An autonomous land vehicle in a neural network, in *Advances in Neural Information Processing*, ed. D. Touretzky, Vol. 1, (Morgan-Kaufmann Publishers, San Mateo, CA, 1989), pp. 305–313.
36. M. Rosenblum and L. S. Davis, An improved radial basis function network for visual autonomous road following. *IEEE Trans. Neural Networks* **7**(5), 1111–1120 (1996).
37. D. Roy and A. Pentland, Learning words from sights and sounds: A computational model, *Cogn. Sci.* **26**(1), 113–146 (2002).
38. A. J. Rudnicky, A. G. Hauptmann and K. F. Lee, Survey of current speech technology, *Commun. ACM* **37**(3), 52–57 (1994).
39. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Prentice-Hall, Upper Saddle River, New Jersey, 1995).
40. M. Sidman and W. Tailby, Conditional discrimination versus matching to sample: An expansion of the testing paradigm, *J. Exp. Anal. Behavior* **37**, 5–22 (1982).
41. R. S. Sutton and A. Barto, *Reinforcement Learning* (MIT Press, Cambridge, Massachusetts, 1998).
42. M. Tomasello, The role of joint attentional processes in early language development, *Language Sci.* **10**, 69–88 (1988).
43. D. S. Touretzky and L. M. Saksida, Operant conditioning in skinnerbots, *Adaptive Behaviors* **5**(3 & 4), 219–247 (1997).
44. J. K. Tsotsos, Analyzing vision at the complexity level, *Behavioral and Brain Sciences* **13**, 423–469 (1990).
45. J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis and F. Nuflo, Modeling visual attention via selective tuning, *Artif. Intell.* **78**, 507–545 (1995).
46. A. M. Turing, Computing machinery and intelligence, *Mind* **59**, 433–460 (1950).
47. L. S. Vygotsky, *Thought and Language* (MIT Press, Cambridge, Massachusetts, 1962), trans. E. Hanfmann & G. Vakar.
48. A. Waibel and K. Lee, *Readings in Speech Recognition*. Morgan Kaufmann, San Mateo, CA, 1990.
49. I. Wallace, D. Klahr and K. Bluff, A self-modifying production system of cognitive development, in *Production System Models of Learning and Development*, eds. D. Klahr, P. Langley and R. Neches (MIT Press, Cambridge, Massachusetts, 1987), pp. 359–435.
50. C. Watkins, Q-learning, *Artif. Intell.* **8**, 55–67 (1992).
51. J. Weng, Learning in image analysis and beyond: Development, in *Visual Communication and Image Processing*, eds. C. W. Chen and Y. Q. Zhang (Marcel Dekker, New York, 1998). A revised version from “Living Machine Initiative,” MSU CPS Tech. Report CPS-96-60, 1996, pp. 431–487.
52. J. Weng, N. Ahuja and T. S. Huang, Learning recognition and segmentation using the Cresceptron, *Int. J. Computer Vision* **25**(2), 109–143 (1997).
53. J. Weng and S. Chen, Vision-guided navigation using SHOSLIF, *Neural Networks* **11**, 1511–1529 (1998).
54. J. Weng and W. Hwang, Online image classification using IHDR, *Int. J. Document Analysis and Recognition* **5**(2–3), 118–125 (2002).
55. J. Weng, W. S. Hwang, Y. Zhang and C. Evans, Developmental robots: Theory, method and experimental results, in *Proc. 2nd Int. Conf. Humanoid Robots*, Tokyo, Japan, October 8–9 (IEEE Press, 1999), pp. 57–64.

56. J. Weng, W. S. Hwang, Y. Zhang, C. Yang and R. Smith, Developmental humanoids: Humanoids that develop skills automatically, in *Proc. First IEEE Conf. Humanoid Robots* (MIT Press, Cambridge, Massachusetts, September 7–8, 2000).
57. J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur and E. Thelen, Autonomous mental development by robots and animals, *Science* **291**(5504), 599–600 (2001).
58. J. Weng, Y. Zhang and Y. Chen, Developing early senses about the world: ‘object permanence’ and visuoauditory real-time learning, in *Proc. Int. Joint Conf. Neural Networks*, Portland, Oregon, July 20–24, 2003.
59. J. Weng, Y. Zhang and W. Hwang, Teaching a learning vehicle — A developmental perspective, in *Proc. Robotics and Mechatronics Congress (RMC2001)*, Singapore, June 6–8, 2001.
60. J. Weng, Y. Zhang and W. Hwang, Candid covariance-free incremental principal component analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence* **25**(8), 1034–1040 (2003).
61. D. S. Wheeler, A. H. Fagg and R. A. Grupen, Learning prospective pick and place behavior, in *Proc. 2nd IEEE Int. Conf. Development and Learning* (MIT Press, Cambridge, Massachusetts, June 12–15, 2002).
62. D. J. Wood, J. S. Bruner and G. Ross, The role of tutoring in problem-solving, *Journal of Child Psychology and Psychiatry*, 89–100 (1976).
63. S. Zeng, D. Cherba and J. Weng, Dav developmental humanoid, in *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM 2003)*, Kobe, Japan, July 20–24, 2003, pp. 974–980.
64. S. Zeng and J. Weng, Obstacle avoidance through incremental learning with attention selection, in *Proc. IEEE Conf. Robotics and Automation*, New Orleans, Louisiana, April 26–May 1, 2004.
65. N. Zhang and J. Weng, A developing sensory mapping for robots, in *Proc. IEEE 2nd Int. Conf. Development and Learning (ICDL 2002)* (MIT Press, Cambridge, Massachusetts, June 12–15, 2002), pp. 13–20.
66. N. Zhang, J. Weng and X. Huang, Progress in outdoor navigation by the SAIL developmental robot, in *Proc. SPIE Int. Symp. Intelligent Systems and Advanced Manufacturing*, Vol. 4573, Newton, Massachusetts, October 28–November 2, 2001.
67. Y. Zhang and J. Weng, Grounded auditory development by a developmental robot, in *Proc. INNS-IEEE Int. Joint Conf. Neural Networks*, Washington, DC, July 14–19, 2001, pp. 1059–1064.
68. Y. Zhang and J. Weng, Action chaining by a developmental robot with a value system, in *Proc. IEEE 2nd Int. Conf. Development and Learning (ICDL 2002)* (MIT Press, Cambridge, Massachusetts, June 12–15, 2002), pp. 53–60.
69. Y. Zhang and J. Weng, Conjunctive visual and auditory development via real-time dialogue, in *Proc. 3rd Int. Workshop Epigenetic Robotics*, Boston, Massachusetts, August 4–5, 2003, pp. 974–980.



Juyang (John) Weng received his Ph.D. degree in Computer Science from University of Illinois, Urbana, IL, USA, in January 1989. He currently is a Professor at the Department of Computer Science and Engineering, Michigan State University, USA. His research interests include computer vision, speech recognition, human-machine multimodal interface using vision, audition, speech, gesture and actions, and intelligent robots. He is the author of over one hundred research articles and book chapters. He is a co-author (with T. S. Huang and N. Ahuja) of the book *Motion and Structure from Image Sequences* (Springer-Verlag, 1993). He is an editor-in-chief of *International Journal of Humanoid Robotics* and an associate editor of *IEEE Trans. Pattern Recognition and Machine Intelligence*. He is the chairman of the *Autonomous Mental Development Technical Committee of the IEEE Neural Networks Society*. He was an associate editor of *IEEE Trans. Image Processing* (1994–1997), a program co-chair of the *NSF/DARPA Workshop on Development and Learning (WDL)*, held April, 5–7, 2000 at Michigan State University (www.cse.msu.edu/dl/), and a program co-chair of the *IEEE 2nd International Conference on Development and Learning (ICDL'02)*, held at Massachusetts Institute of Technology, Cambridge, MA, USA, June 12–15, 2002 (www.egr.msu.edu/icdl02/). His home page is: www.cse.msu.edu/~weng/.