This article was featured in NNS coNNectionS, volume 1, number 4, pp. 8-13, nov. 2003.

Feature Article

Juyang Weng
Michigan State University

Autonomous Mental Development:
A New Frontier for Computational Intelligence

Juyang Weng
Department of Computer Science and Engineering
Michigan State University, East Lansing, MI 48824, USA

***Abstract*. Approaches to computational intelligence, including neural networks, fuzzy systems and evolutionary computation, are converging to a common frontier --- autonomous mental development (AMD). This article explains what AMD is and why computational intelligence can fully expand its power at this frontier. As an example, this paper discusses a theory, an architecture, and some experimental results of AMD.**

1. Introduction

The realization of an intelligent species is accomplished through evolution. Between generations, the genome is evolved. Within a generation, each genome (the genotype) is developed into an individual (the phenotype), through interactions with the environment. Cross-generation modification of the genome is realized through development, mating and reproduction.

The development of a human being starts from a single cell --- a fertilized egg, called zygote. The genome in its nucleus contain all the genetic information necessary for developing a human individual in the human environments. At the conception time, a genome program starts to run. We call it a *developmental program*. The program "computes" biologically, through interactions with the environment, to accomplish two types of development: (1) the body (physical) development and (2) the mental development. Through the former, a single-cell zygote develops into a multi-pound body of a newborn at the birth time, and further into a big and strong adult body. Through the latter, a human mind is developed which is able to sense, understand, and act intelligently on the environment, at a competence level of the corresponding age group.

Biologically, these two types of development are closely related to each other. Without the body development, mental development is not possible. Without the mental development, the body development can not survive the competitive environment.

In his pioneering paper published in 1950 titled "Computing Machinery and Intelligence" (Turing, 1950), Alan Turing envisioned a machine that can learn, which he called "child machine." He wrote:

> "Our hope is that there is so little mechanism in the child brain that something like it can be easily programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child."

A large amount of studies on machine learning was motivated by human learning, but not necessarily development per se. Examples include Soar (Laird et al., 1987), ACT-R (Anderson, 1993), Hidden Markov Models (HMM), Markov Decision Process (MDP). They are computational models, not a model generator.

Some studies did intend to model cognitive development, but in a symbolic way. Examples include (Wallance et al., 1987) and (Drescher, 1991). Since symbols are designed by humans, symbolic representation is not suited for automatic generation of new, unknown representation as required by AMD (Weng, 2002).

Numeric incremental learning has been a major characteristic of the computational intelligence (McCllelland et al., 1986). There are some systems that automatically generate classification networks from sensory experience, e.g., Cresceptron (Weng et al., 1997), SHOSLIF (Hwang and Weng, 1997).

All of the above efforts aim at simulating some aspects of learning and development. However, they did not intend to accomplish autonomous mental development. For example, it is a human who manually maps a computational framework to a given task. This type of mapping is called *task-to-representation mapping*. The way a human manually establishes such a mapping takes several forms:

1. Mapping between real-task concepts and symbols (e.g., soar and ACT-R).

2. Mapping between real-task concepts and model nodes (e.g., FSM, HMM and MDP).

3. Mapping between real-task features and input terminals (e.g., skin color and region position as input terminals for detecting human faces).

AMD does not require a human being to provide any of the above task-to-representation mapping.

The autonomous mental development did not receive serious attention in the research community, until the late 1990s when the SAIL (Self-Organizing Autonomous Incremental Learner) robot (Weng, 1998, Weng et al., 1999) and the Darwin V robot (Almassy et al., 1998) conducted real-time online experiments on autonomous cognitive development. A 2001 article in *Science* (Weng et al., 2001) summarized the pivotal role that mental development can play in machine intelligence.

What is autonomous mental development? What are the basic differences between traditional machine learning and autonomous mental development? How techniques developed in the computational intelligence community can be further developed or modified to accomplish autonomous mental development?

This article discusses these issues. A new agent model required for AMD is described in Section 2. Section 3 presents the paradigm of autonomous mental development. Section 4 introduces an example of software architecture that is capable of autonomous mental development. Section 5 discusses the type of representation suited for autonomous mental development. Section 6 outlines some experimental results with the SAIL and Dav developmental robots. Section 7 provides concluding remarks.

2. AMD Needs a New Kind of Agent

Defined in the standard AI literature[1], an agent is something that senses and acts.

[1] - See, e.g., an excellent textbook by Russell & Norvig (Russell and Norvig, 1995) and an excellent survey by Franklin (Franklin and Graesser, 1997).

The environment ***E*** of an agent is the world outside the agent. It is true that ***E*** should include the agent body. However this is still not sufficient. The target of perception must include the brain itself (of course, not all the brain).

We define the internal environment of an agent to be the brain of the agent (or its central nervous system). All the rest of the world, including the agent body and the world around it correspond to the external environment. Note that the agent body belongs to the external environment, not internal.

This model is for an agent that perceives only the external environment and acts on only the external environment. This well accepted model 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 the agent to autonomously act on.

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., (Kandel et al., 2000), pp. 396 - 403). AMD requires a new model of agent, called *self-aware and self-effecting* (SASE) agent.

For example, attention selection and action release are internal actions and the sense of these actions are internal senses.

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 able to understand what it is doing and why. In other words, it is not self-aware. Further, the behaviors that it generates are for the external world only, not for the brain itself. Thus, it is not able to autonomously change its internal decision steps.

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

3. Autonomous Development Paradigm

The traditional paradigm for developing human engineered systems is manual.

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 and the automatic execution. 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***. 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 an agent **A**:

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

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

3.2. Autonomous development

The autonomous development paradigm has two phases, (1) construction and programming phase and (2) the autonomous development phase.

In the first phase, tasks that the agent will end up learning are unknown to the robot programmer. The programmer might speculate some possible tasks, but writing a task-specific representation is not possible without actually being given a task. The ecological conditions under which the robot will operate, e.g., land-based or underseas, are provided to the human constructor so that he can design the agent body appropriately. He writes a *task-nonspecific* program called a *developmental program*, which controls the process of mental development. Thus, the newborn agent $A(t)$ at time $t=0$ is a function of a set of ecological conditions only, but not the task:

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

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

When 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. This phase is called autonomous development phase. Human teachers can affect the developing robot only as a part of the environment, through the robot's sensors and effectors. After the birth, the internal representation is not accessible (closed) to the human teacher.

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

4. An Architecture Example

A key architecture for development is the sensorimotor system. When co-developed with multiple sensorimotor systems, an integrated network of sensorimotor system has the potential to deal with high-level cognitive behaviors such as abstract reasoning and planning.

4.1. Overview

Fig. 1 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 systems mediated by subsumption. Each sensorimotor system handles a complete mapping from the sensory input all the way to the motor output. The central pathway from the retina, through sensory mapping and cognitive mapping to motor mapping is a major sensorimotor system. The innate behavior pathway (on the right in the figure) is another but simpler sensorimotor system.

These two systems are mediated by the subsumption in the motor mapping. The subsumption module mediates the action outputs from different sources so that the action from a pathway that is positioned higher in the sensory integration hierarchy has a higher priority (Brooks, 1986).

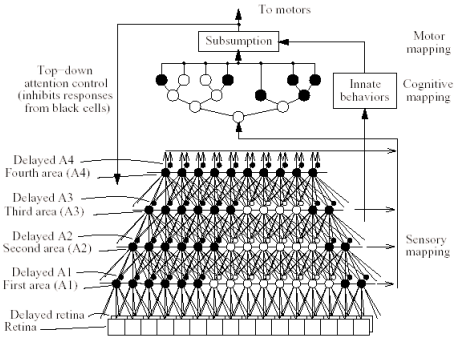

Figure 1: The flow diagram of a developmental vision system. The responses from black cells are inhibited by the attention control but those from the white cells are passed at this time instance. The sensorimotor system for innate behavior is illustrated by a block only for simplicity, but it also contains the three mappings which are, however, much smaller in storage and simpler in the mappings it realizes.

A sensory mapping has two major functions: (1) It provides parallel responses for all possible receptive fields which are fed into the following cognitive mapping for learning. (2) 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 (for feature detection) is hand-designed but is further refined while the connection weights are incrementally computed (learned or developed) from sensing experience, using PCA, ICA, etc. We used the Candid Covariance-free Incremental Principal Component Analysis (CCIPCA) (Weng et al., 2003). More detail about sensory mapping is available in (Zhang and Weng, 2002a).

A cognitive mapping incrementally 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 (to be explained below). Incremental Hierarchical Discriminant Regression (IHDR) (Hwang and Weng, 2000, Weng and Hwang, 2002) is used to self-organize the input space of f into a hierarchy of (nested) partitions organized into a tree structure. Each cell in a coarse partition is refined by a fine partition in the next finer level in the tree. In each node, the space is represented by its own automatically developed most-discriminating feature subspace, in which the boundary of cells of finer partition is determined by Bayesian estimation. This results in a quasi-optimal generalization boundary, conditioned on the current coarseness of the partition.

Such recursive coarse-to-fine partitions end 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 being 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, making it possible to achieve real-time speed even when the number of context prototypes is very large. It has been systematically demonstrated (Hwang and Weng, 2000) that HDR out-performs many well-known classifiers, including the support vector machines, in a series of high-dimensional tests.

The architecture shown in Fig. 1 is equally well applicable to other sensory modalities, such as vision, speech and touch, each with a different set of developmental parameters (e.g., the extent of temporal context). This is practical because representation (e.g., feature detectors using PCA and tree structure using HDR) is generated automatically from the sensing experience of that sensing modality.

4.2. Contexts

A sensorimotor system is a predictor and a doer. At each time instant t, a sensorimotor system has the last context:

$$l(t) = (x_l(t), a_l(t)) \quad (3)$$

which contains the *last sensation* $x_l(t)$ and the *last action* $a_l(t)$. A sensorimotor system needs to predict future sensations and actions. We call them the *primed sensation* x_p and the *primed action* a_p, respectively. They form what is called the *primed context*

$$p(t) = (x_p(t), a_p(t)). \quad (4)$$

Let $\mathcal{P}$ denote the space of the primed contexts. Therefore, we have defined four types of context information. They are positioned in the input and output spaces and along the time axis.

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 $p_1(t)$, ... , $p_k(t)$ (e.g., left and right turns at a Y junction). This mapping is accomplished by a particular cognitive mapping called reality mapping R:

$$\{p_1(t), \ldots, p_k(t)\} = R(l(t)). \quad (5)$$

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

$$R: \mathcal{L} \mapsto 2^{\mathcal{P}} \quad (6)$$

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:

$$V(R(l(t))) = V(\{p_1(t), p_2(t), \ldots, p_k(t)\}) = p_i(t). \quad (7)$$

where $1 \le i \le k$ 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 the space of $\mathcal{P}$:

$$V: 2^{\mathcal{P}} \mapsto \mathcal{P}. \quad (8)$$

Further, a single mapping R is not sufficient. We need multiple ones, R maps to near future and F maps to far future.

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 $\mathcal{P}$

$$\mathcal{L} \overset{R,F}{\mapsto} 2^{\mathcal{P}} \overset{V}{\mapsto} \mathcal{P}$$

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

4.3. Architecture view

The coarse architecture of a developmental machine is designed, which needs to be included in the genomic representation in *evolutionary computation*. The fine architecture and networks are developed (grown) incrementally from experience. Thus, *neural network* ideas can be used in AMD. Due to the high-dimensional context space and the system must work while it "builds" itself, many methods that deal with uncertainty, including *fuzzy systems*, can be used to enable better generalization under a finite amount of developmental experience.

A block diagram of the architecture of a sensorimotor system is shown in Fig. 2.

As shown in Fig. 2, 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

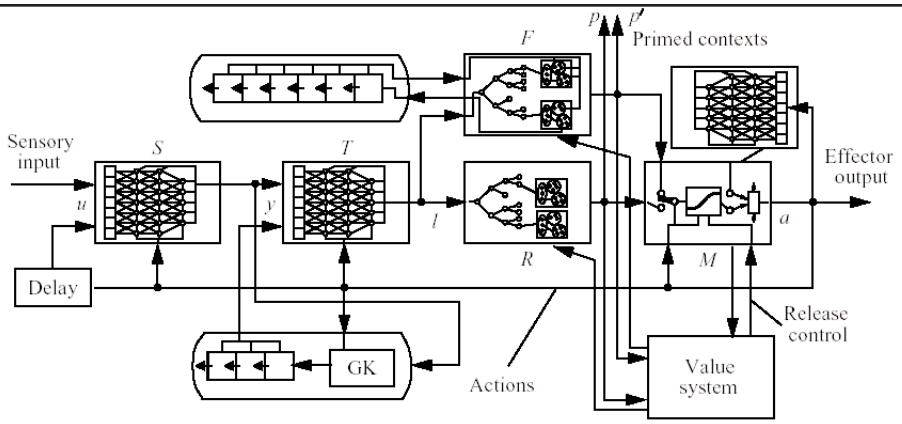

Figure 2: A block diagram of the architecture of a sensorimotor subsystem. ***S*** is a spatial sensory mapping; ***T*** is a spatiotemporal sensory mapping. GK: gate keeper, an internal effector to actively control the update of the last context. ***R*** the reality mapping and ***F*** the priming mapping, both are implemented by the cognitive mapping 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.

in Fig. 2, is its target for perception and cognition.

A sensory mapping is needed wherever an attention selection effector or dimension reduction is needed. As shown in Fig. 2, 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 both space and time.

The priming mapping ***F***, implemented by a cognitive mapping, needs a *prototype updating queue*, whose function is to update the primed context using Q-learning algorithm (Watkins, 1992) in a recursive way. The reality mapping does not need such a queue because it only predicts the next step.

The motor mapping ***M*** of a sensorimotor system generates concise representation for stereotyped actions. 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. 2. The new part of the motor 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 developmental machine is also allowed to perform autonomous learning. The innate value system (Huang andWeng, 2002) 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 understanding of social norms.

5. Internal Representation

The term "internal representation" refers to the representation used internally by the agent. For a human or developmental robot, it is the "brain" or central nervous system.

5.1. World and mind concepts

In the current AI literature, the distinction between world concepts and mind concepts have been largely ignored. A main reason is that it is the human programmer to design a representation (e.g., logic based representations, Soar, ACT-R, HMM and MDP) 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 clean problem. Algorithms that perform operations on such a world representation are largely hand-designed. These hand-designed algorithms applied to a world-center symbolic representation do not provide a possibility for machine to "step back" to examine and understand what it is doing and fundamentally change the way resolutions are reached.

When we discuss representation, it is important to distinguish between the actual physical world and the mental effects that it causes.

Definition 1: *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[2] *of an agent is internal with respect to the nervous system (including the brain) of the agent.*

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 the fact of a part of the world, no matter we call 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 2: *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. All these prototypes correspond to the same human face.

The world is the *best* representation of the world in terms of authenticity. A programmer may generate an artificial representation about a room, using some data structure. It is a world-centered representation, which is never as good as the world itself in authenticity, but might be easier to use by his program.

A mind-centered representation is specific to a particular agent (mind). It can only represent mind concepts. 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 and can be an illusion or totally false.

5.2. Symbolic and numeric representations

[2] *- The 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.*

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 $\boldsymbol{v = (v_1, v_2, ..., v_n)}$ where $\boldsymbol{v}$ (optional) is the name token of the object and $\boldsymbol{v_1, v_2, ..., v_n}$ 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. The set of attributes is unique in the sense, e.g., that the object's weight is given by the unique entry $\boldsymbol{v_1}$.

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

1. Each component in the representation has a predefined meaning about the object in the external world.

2. Each attribute is represented by a unique variable in the representation.

3. The representation is unique for a single corresponding physical object in the external environment.

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 (agent) concept to use a symbolic representation. In fact, the distinction between a world-centered concept and a mind-centered representation has not been emphasized, since they typically have a one-to-one correspondence. The former is approximated by the latter. For example, this has been a very common practice in robotics where a sensed 3-D range map is represented by a data structure. However, for developmental robots, it is not possible to use symbolic representation for mind-centered concepts, as explained below. We have been using numeric representation for developmental robots.

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. It is in a vector form $\mathbf{v = (v_1, v_2, ..., v_n)}$, where $\boldsymbol{v}$ (optional) denotes the vector and $\boldsymbol{v_i}$, $\boldsymbol{i = 1, 2, ..., n}$ corresponds to either a sensory element (e.g., pixel or receptor) in the sensory input, a motor control terminal in the action output, or a function of them.

For example, suppose that an image produced by a digital camera is denoted by a column vector $\boldsymbol{x}$, whose dimension is equal to the number of pixels in the digital image. Then $\boldsymbol{x}$ is a mind-centered numeric representation, and so is $\boldsymbol{f(x)}$ where $\boldsymbol{f}$ is any function. A numeric representation of dimension $\boldsymbol{n}$ can represent the response of $\boldsymbol{n}$ neurons.

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

Figure 3: The SAIL robot (left) and Dav robot (right).

High dimension in a numeric representation implies that the representation is distributed over many numbers, $\boldsymbol{v_1, v_2, ..., v_n}$, where each number $\mathbf{v_i}$, $\boldsymbol{i = 1, 2, ..., n}$, corresponds to a component in the vector. Mind centered numeric representation in AMD implies that AMD is a frontier that computational intelligence approaches are suited for.

6. Experiments

6.1. Developmental system projects

Our decade-long effort in enabling machines to grow their perceptual and behavioral capabilities has gone through four systems: Cresceptron (1991 - 1995), SHOSLIF (1993 - 2000), SAIL (1996 - present) and Dav (1999 - present). Initial tests of the developmental program were conducted in simulation, where the ground truth is available and the system perception and behaviors can be preciously recorded and analyzed. When the simulation was successful, the developmental program was moved to real robots for real-time, online tests. Here, we discuss some of the experiments with the SAIL and Dav robots shown in Fig. 3.

Three types of learning modes have been implemented on SAIL with the SAIL-3 developmental program: learning by imitation (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 online in real-time, except those stated otherwise. Movies are available at http://www.egr.msu.edu/~weng/research/LM.html.

6.2. Attention selection

We have designed and implemented a general-purpose sensory mapping, called "Staggered Hierarchical Mapping (SHM)" and its developmental algorithm (Zhang and Weng, 2002a). It is a biologically motivated (but simplified) model of the early visual pathway. It enables a robot to select attention and recognize objects from occluded views.

A human teacher taught the robot by taking it for a walk along the corridors of MSU's Engineering Building. The human programmer does not design any of the features. Instead the developmental program automatically develops feature detectors from the scenes observed. After a few trips along slightly different trajectories along the corridors, the human teacher started to let the robot "go free." Experimental results have shown that the SAIL robot can navigate autonomously

using its vision-based sensorimotor skills that were acquired through online, real-time developmental learning (Weng et al., 1999). It perceived the scenes from its video cameras without using any range sensor.

The developmental speech learning demonstrated (Zhang and Weng, 2001) is very different from the traditional speech learning: (1) The continuous auditory streams have not been segmented and labeled (thus, autonomous learning is possible). (2) 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 that it is designed to recognize (e.g., various utterances of the word ``good''). (3) No syntax is involved during programming (e.g., the system can learn multiple languages concurrently). The robot performs autonomous language acquisition.

6.3. Scaffolding: transfer and chaining

The objective of this project is to enable a robot to develop complex skills after the acquisition of simple ones (Zhang and Weng, 2002b). The robot is able to transfer multiple cognitive and behavioral skills learned in a setting to new settings and can chain them by taking into account new contexts. Upon learning the basic gripper tip movements (drawing an individual petal), the SAIL robot learned to combine individually instructed movements to be a composite one (drawing a flower) invoked by a single verbal command without any reprogramming.

6.4. Range-based collision avoidance

The Dav robot learned to avoid collisions using its Sick laser range scanner, through an online incremental learning process. To reduce the amount of interactive training needed, an attention mechanism is used to suppress some parts of the range vector that the robot does not need to pay attention to. Nearby objects are attended and faraway ones are not, unless there is no nearby object.

Experimental results showed that Dav can reliably navigate in dynamic changing environments without hitting obstacles, static or people (Zeng and Weng, 2003).

7. Conclusions

Computational intelligence has yet to demonstrate high level reasoning and multimodal understanding. Autonomous mental development is a natural way for it to demonstrate its power. The numeric computation nature of the AMD architecture example indicates that computational intelligence has a higher potential than many have realized so far.

References

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

Anderson, J. R. (1993). *Rules of the Mind*. Lawrence Erlbaum, Mahwah, New Jersey.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14-23.

Drescher, G. L. (1991). *Made-Up Minds*. MIT Press, Cambridge MA.

Franklin, S. and Graesser, A. (1997). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent Agents III, Lecture Notes on Artificial Intelligence*, pages 21-35, Berlin. Springer-Verlag.

Huang, X. and Weng, J. (2002). Novelty and reinforcement learning in the value system of developmental robots. In *Proc. Second International Workshop on in Robotic Systems (EPIROB'02)*, pages 47- 55, Edinburgh, Scotland.

Hwang, W. and Weng, J. (1997). Vision-guilded robot manipulator control as learning and recall using SHOSLIF. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, Albuquerque, NM.

Hwang, W. S. and Weng, J. (2000). Hierarchical discriminant regression. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1277-1293.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M., (Eds.) (2000). *Principles of Neural Science*. McGraw-Hill, New York, NY, 4th edition.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1-64.

McClelland, J. L., Rumelhart, D. E., and The PDP Research Group, (Eds.) (1986). *Parallel Distributed Processing*, volume Vol. 1 and Vol. 2. MIT Press, Cambridge, MA.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Upper Saddle

River, NJ. Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59:433-460.

Wallance, I., Klahr, D., and Bluff, K. (1987). A selfmodifying production system of cognitive development. In Klahr, D., Langley, P., and Neches, R., (Eds.), *Production System Models of Learning and Development*, pages 359-435. MIT Press, Cambridge, MA.

Watkins, C. (1992). Q-learning. *Artificial Intelligence*, 8:55-67.

Weng, J. (1998). Learning in image analysis and beyond: Development. In Chen, C. W. and Zhang, Y. Q., (Eds.), *Visual Communication and Image Processing*, pages 431 - 487. Marcel Dekker, New York, NY. A revised version from "Living Machine Initiative," MSU CPS Tech. Report CPS-96-60, 1996.

Weng, J. (2002). A theory for mentally developing robots. In *Proc. IEEE 2nd International Conference on Development and Learning (ICDL 2002)*, pages 131-140, MIT, Cambridge, MA.

Weng, J., Ahuja, N., and Huang, T. S. (1997). Learning recognition and segmentation using the Cresceptron. *International Journal of Computer Vision*, 25(2):109-143.

Weng, J. and Hwang, W. (2002). Online image classification using IHDR. *International Journal on Document Analysis and Recognition*, 5(2-3):118-125.

Weng, J., Hwang, W. S., Zhang, Y., and Evans, C. (1999). Developmental robots: Theory, method and experimental results. In *Proc. 2nd International Conference on Humanoid Robots*, pages 57-64, Tokyo, Japan. IEEE Press.

Weng, J., Hwang, W. S., Zhang, Y., Yang, C., and Smith, R. (2000). Developmental humanoids: Humanoids that develop skills automatically. In *Proc. First IEEE Conf. on Humanoid Robots*, MIT, Cambridge, MA. IEEE Press.

Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*, 291(5504):599-600.

Weng, J., Zhang, Y., and Hwang, W. (2003). Candid covariance-free incremental principal component analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(8):1034-1040.

Zeng, S. and Weng, J. (2003). Online-learning and attention-based approach to obstacle avoidance behavior. Technical Report MSU-CSE-03-26, Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan.

Zhang, N. and Weng, J. (2002a). A developing sensory mapping for robots. In *Proc. IEEE 2nd International Conference on Development and Learning (ICDL 2002)*, pages 13-20, MIT, Cambridge, MA.

Zhang, Y. and Weng, J. (2001). Grounded auditory development by a developmental robot. In *Proc. INNS-IEEE International Joint Conference on Neural Networks*, pages 1059-1064,Washington, DC.

Zhang, Y. and Weng, J. (2002b). Action chaining by a developmental robot with a value system. In *Proc. IEEE 2nd International Conference on Development and Learning (ICDL 2002)*, pages 53-60, MIT, Cambridge, MA.

Juyang Weng received his PhD degree in Computer Science from University of Illinois, Urbana, IL USA, January 1989. He is a professor at the Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan, 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 coauthor (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, an associate editor of IEEE Trans. on Pattern Recognition and Machine Intelligence. He was an associate editor of IEEE Trans. on 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 (MSU), East Lansing, MI (http://www.cse.msu.edu/dl/), and a program cochair of the IEEE 2nd International Conference on Development and Learning (ICDL'02), held at Massachusetts Institute of Technology, Cambridge, MA, June 12- 15, 2002 (http://www.egr.msu.edu/icdl02/).