

The Developmental Approach to Intelligent Robots

John J. Weng

Department of Computer Science
Michigan State University
East Lansing, MI 48824
weng@cps.msu.edu

Abstract

Integration of major cognitive capabilities and behavioral capabilities is crucial for robots to perform harder, general tasks. Several requirements are raised in this paper: (1) a successful integration of such capabilities need to start with an architecture that is suited for integration, (2) the development of each individual capability must be an integral process of overall integration process, and (3) the content-level integration process must be fully automated. The developmental approach to intelligent robots is presented in this paper as a way for integrating cognitive and behavioral capabilities in an automated way. This approach is motivated by human cognitive development from infant to adult. Central in the approach is the methodology for sensor-effector-rich robots to perform general-purpose, autonomous, incremental learning directly using their sensors and effectors through interactions with the real-world environment. In other words, the objective is to fully automate the learning process, so that the robots can learn in an automatic mode that is close to the way animals (and humans) learn. A comparison between the developmental approach and other existing approaches is provided.

Introduction

Existing works on robotics research have produced a rich collection of methods and robotic systems. Some have demonstrated impressive capabilities. However, these studies tend to concentrate on a narrowly defined scenario with a specialized capability.

Requirement of cognition

The capability of a robot to perform harder, general tasks depends very much on its cognitive and behavioral capabilities. For example, if a robot has only range sensing capability, it cannot navigate along outdoor walkways of a campus, since range sensors cannot tell much difference among road and nonroad surfaces, such as concrete blocks, asphalt, grass, and soil. In an indoor environment, it is difficult for such a robot to identify a closed door that it must open and pass through. A robot with visual and auditory capabilities

will be able to handle more general scenarios and may interact with its environments, including humans, using the visual and auditory modalities. Tasks that require manipulatory behaviors, such as pouring a cup of water into another, also require sophisticated coordination between sensors and effectors.

Four existing approaches

However, integration of advanced sensors and effectors faces a grim reality: the complexity of the nature of integration quickly reaches a point that is impractical to handle by the current existing approaches. These approaches can be divided into the following four categories:

1. The world-knowledge-based approaches. The approaches in the category typically require a predefined problem space or world space. Researchers in each problem area have been manually developing knowledge-level theories and methods, and using them to write programs or build hardware. Then, they manually “spoon feed” knowledge into the systems at the programming level (e.g., CART (Moravec 1982), CYC (Lenat 1995) (Lenat, Miller, & Yokoi 1995), lexical database, WordNet (Miller 1995)). Such an approach may produce a system that appears to produce some sophisticated results. However, the limitations of such systems have been recognized (Brooks 1991) (Franklin 1997). Such a methodology requires a huge amount of human labor and it faces a fundamental limit of human labor in adequately modeling and specifying the cognitive process required by challenging robotic tasks.
2. The behavior-based approaches. Approaches of this type avoid modeling the world and instead they model robot behavior. The subsumption architecture was proposed by Brooks to allow a more sophisticated behavior layer to be added to the existing primitive behavior layers (Brooks 1986). Each layer is a finite state machine, with states defined and named by the programmer. The programmer is also responsible to program the state machine in each layer for each desired behavior. Thus, this approach can be characterized by the terms “manually-modeling-behavior

and hand-coding-behavior”. Aloimonos (Aloimonos 1990) and others also advocated behavior-based approach for active vision.

3. The evolutionary approaches. This approaches is motivated by evolution of biological species. The law of survival of the fittest is used to select advantageous genotypes which code the structure and/or behavior of simulated robots (Mataric 1990) (Harvey 1992). So far, the selection process have been mostly simulated by computers using a simulated environment, due to the obvious difficulties in carrying on evolutionary process with a large number of robots and performing long time physical evolution. The simulation method is attractive due to the low cost benefit. Also, it does not require the programmer to code knowledge or behavior rules. However, evolutionary approaches leave the hard task of intelligent robot design to the process of random trials and environment selection. The computation time required to find such a design and the cost of real-robot evolution are two fundamental difficulties the evolutionary approaches face, which are especially great when high-dimensional perception and cognition are required, such as vision, speech and language.
4. The learning approaches. This category includes all the learning methods, such as supervised learning and reinforcement learning. Learning approaches are typically more efficient than the corresponding evolutionary approaches. This is because, with the former, the learning mechanism of the system is hand-coded by the programmer, but with the latter, the mechanism is either absent or has to emerge from the trial-and-selection process. For high-dimensional input, learning seems the only viable method. Various learning methods have produced impressive results for challenging cognition tasks involving complex modalities, such as visual recognition (e.g.,(Turk & Pentland 1991) (Swets & Weng 1996b)), speech recognition (e.g., using HMM (Rabiner, Wilpon, & Soong 1989) (J. R. Deller, Proakis, & Hansen 1993)), vision-guided robot manipulation (e.g., (Hwang, Howden, & Weng 1996)) and vision-guided navigation (e.g.,(Weng & Chen 1996)). However, a huge amount of manual labor required in training such a system hinders further scaling up to robotics tasks that are more general, of larger-size, and harder. A major reason is that the machine learning process is not automated.

In actuality, a particular system may use a combination of several approaches. For example, Robot-Soar (Laird *et al.* 1991) combines a world-knowledge-based approach with a learning approach. It requires the human to feed knowledge about the environment and to define the problem space. Then, the system learns to perform predefined tasks.

The developmental approach

This article proposes a new approach: the developmental approach. It is rooted in human cognitive development from infant to adult. The renowned developmental psychologist Jean Piaget (Gruber & Voneche 1977) (Carey 1990) (Carey 1985) divided human cognitive development roughly into four major stages, as summarized in Table 1. There is no doubt that these

Table 1: Piagetian Stages of Human Cognitive Development

Stage	Ages	Characteristics
Sensorimotor	0 – 2	Not capable of symbolic representation
Preoperational	2 – 6	Egocentric, unable to distinguish appearance from reality; incapable of certain types of logical inference
Concrete operational	6 – 12	Capable of the logic of classification and linear ordering
Formal operational	12 –	Capable of formal, deductive, logic reasoning

four stages have a lot to do with neural development in the brain. From infant to adult, while each human individual undergoes physical development to become taller and stronger, he or she also goes through its cognitive development which is made possible through extensive interactions between internal changes (e.g., neural growth) and external environmental changes (e.g., the effect of an action).

A central task for the developmental approach is to realize AA-learning¹ — an automated learning mode that is close to the way in which humans (and animals) learn. Human interaction with the environment is a dynamic, autonomous, online, incremental, continuous learning process. Thus, the developmental approach requires humans to hand-code mechanism of automated learning, instead of replying on the evolution process. Since the learning process with AA-learning is fully automatic, integration of various sensors and effectors is realized automatically with the developmental approach.

What is AA-Learning?

An examination of unautomated learning can help the understanding of AA-learning.

Unautomated learning

Developing a system using unautomated learning approach involves two major stages:

¹AA-learning is named after *automated*, *animal-like* learning without claiming to be complete. The term “automated learning” has been used sometimes just to mean “using computers” where the learning process is not fully automated.

Content-level programming.

The content-level programming here means the programming task for modeling the content or knowledge represented in the information of sensory input, other than preprocessing such as Fourier transforms. Modeling visual shape is an example of content-level programming. The human designers must explicitly define the content-level mapping between the sensory input and the model. Here are some examples. For face recognition, one may design 3 neural networks, one for each of the three facial parts, eyes, nose and mouth, respectively, and another for combining the output of the three neural networks. For speech recognition, a separate HMM is dedicated to each word and other HMMs for phrases that use outputs from the word-level networks. For language understanding, a particular syntax model may be designed for a particular syntax.

Data compilation and training.

Next, humans manually compile sensory data, which includes creating scenarios, collecting sensory data, labeling data, segmenting data, ordering data, feeding data, etc. For supervised learning, typically a class label is assigned to each segmented data item. Repeated training and testing are typically conducted before the system can reach the required level of performance.

This mode of system development can be called *unautomated* mode in that the learning process requires intensive human intervention, even though computers are used.

Two major restrictions are direct consequences from such an unautomated mode of system development — low quantity and low quality of information fed into the system. In terms of quantity, the computers are allowed to observe far less environmental variation, contextual variation, and content domain variation than they really need for the tasks that they are assigned for. It is difficult to conduct extensive system training due to the large amount of manual labor that is required in preparing the training data. In terms of quality, these compiled training data are “dead”. They are very much *disconnected* from the environment from which the data arise. The rich meaning of the live sensory experience is degenerated into isolated segments, each being tied to a class label which is meaningless to the system. The lack of environmental context of manually fed training segments make it impossible for machines to learn beyond what is modeled by the content-level programming. Unfortunately, what is modeled by content-level programming is typically insufficient for the challenging tasks at hand, due to our human limitation in understanding and modeling the complex mechanisms of human cognitive processes.

For example, if we require a robot to interact with human beings, the robot must be able to recognize a human face and distinguish it from, e.g., a tree. Recognition of human faces must cope with a wide vari-

ety of variation factors, such as lighting, viewing angle, viewing distance, and facial changes (e.g., expressions, hair styles, eye wears etc). A similar situation is true in speech recognition (e.g., variation in time warping, coarticulation, intonation, age, gender, etc) and language understanding (ambiguity without context, ambiguity without understanding, cultural differences, language differences). Recognition through a sensing modality requires experience far beyond that single sensing modality. For example, Sinha & Poggio’s Clinton-Gore example (Sinha & Poggio 1996) demonstrates that recognition of human individual is heavily influenced by our experience in society². It is extremely difficult, if not impossible for human system designers to adequately model all these factors, to design effective representation and algorithms to use these models, to collect sufficient knowledge data for these models, and to keep these models and knowledge data up to date.

A fundamental way to address these very challenging issues is to investigate how to automate the training process for robots. In other words, we need to develop theory and technology to realize a new mode for machines to interact with humans and environment and to learn to communicate with human through their sensors and effectors — a mode that relieves humans from heavy burden of programming at content-level and allows the systems to intimately interact with its environment. We call it *AA-learning*.

Automated learning

The AA-learning is a mode of automated, general-purpose learning that is close to the way animals (and humans) interact with its environment and learn. Some key points of this type of learning are:

Domain-extensible. Domain-extensible means that the system is not only applicable to an open number of problem domains but can also move onto new problem domains without need for re-programming. A domain-extendable system like a human can continuously switching among deferent domains. For example, while reading a magazine, a human can switch among recognizing human faces, recognizing human genders, recognizing written characters and recognizing other objects. For a domain-extensible system, the system designer cannot use a problem-specific objective function (e.g., in training a robot leg hopper, the objective is fixed — keeping balance). He cannot define a specific problem space for programming either. Existing learning systems that use a general-purpose learning method typically require the system designer to map a problem into the program internal representation (e.g., to map all the possible situations of a problem into a set of manually defined system states, such as in Robo-Soar (Laird *et al.* 1991)).

²The Clinton-Gore example shows that you recognize these two well-known individuals in a picture even when their facial areas are made to be identical.

Sensing and actions. The system will learn from the real-world environment directly using its sensors and effectors, without requiring humans to serve as a feature detector or sensory-input-to-symbol converter. This is in contrast with symbolic methods such as Soar (Newell 1990) (which works in a symbolic domain) and softbots (Terzopoulos, Tu, & Grzeszczuk 1994) (which work only in a simulation world), but common in the vision, speech and robotics community.

Learn autonomously while performing. The system learns autonomously, continuously, incrementally, in real-time, while performing. The learning phase is also the performance phase. Humans are not in the loop of collecting training data. Humans are a part of the environment that the system *continuously* interacts with.

Combined learning. Reinforcement learning (using reward and punishment) and supervised learning (guided actions) are combined. No behavior is hard-wired (or programmed) in, since such hard-wired behavior cannot work well with other behaviors when the system later learns more sophisticated tasks. Some basic behaviors that are typically innate in animals, such as moving forward and simple turning, are taught and memorized through interactive effector guidance (imposition). This allows the system to start with some basic simple behaviors which will facilitate further learning.

Automatic level building. Capabilities for learning stimulus-response association, reasoning and prediction are automatically built up from low level to higher levels. This is in contrast with existing single-level reinforcement learning methods such as Q-learning (Watkins 1989) and R-learning (Schwartz 1993), manually specifying and building behavior layers in the subsumption architecture (Brooks 1991), and human content-level design in level building for speech recognition (e.g., (Rabiner, Wilpon, & Soong 1989) (J. R. Deller, Proakis, & Hansen 1993)).

Real-time requirement while scaling up.

The system must learn while performing in real-time. Thus, a super-low time logarithmic time complexity $O(\log(n))$ is required, where n is the number of cases that are not only learned but also memorized as necessary. This is a highly challenging task for high-dimensional sensory inputs, such as visual input, and large-size problems.

Learning automation does not mean absolutely no human involvement. In principle, a system with the AA-learning capability can learn on its own without human present. However AA-learning typically does need humans to serve as teachers, as a part of the environment. An automobile moves automatically, but it still needs a human to drive it to do something useful. With AA-learning, human teachers the robot by playing with it on-line, instead of by programming and

knowledge-feeding off-line.

In AA-learning, the learner has three types of channels to communicate with its environment, sensors, effectors, and reward receiver. The learner learns autonomously from the environment by sensing the environment through its sensors and acting on the environment through its effectors. We define a physical channel, which include two parts: reward receiver and action imposer. A physical channel is a special control channel that the teacher can use to exercise control over the learner. Human teachers, as a part of the system's environment, affect how the system learns. For example, the human teachers will show different object examples, verbally state the characteristics of the object and then immediately ask questions about the characteristics of the object. The human teacher may encourage the system to act properly using different rewards at the right time. Among all the possible values received from the reward receiver, the learner has different degrees of preference. The action imposer is a special type of sensory input which has a mandatory effect on the corresponding effectors. It can be used to directly enforce the correct action to be executed immediately. Such an imposition of action and delivery of rewards occur also in human learning. For example, manipulating a child's hand to hold a pen when teaching a child how to use a pen. Rewards to a human child can be food, a good test score etc.

Since what to be learned further must depend on the current performance, a system that learns autonomously must also perform autonomously at the same time. This concurrency of learning and performance modes is a very fundamental difference between human learning and the existing conventional machine learning methods. With a machine that can perform AA-learning, the following goals become possible:

1. the learning mechanism must be able to deal with the full generality of the performance environment.
2. the machine is able to improve while performing real tasks.
3. various tasks can be learned without reprogramming.

AA-learning is a new type of learning for machines, although humans and other animals routinely perform this type of learning.

The SAIL Project

The current ongoing SAIL (Self-organizing, Autonomous, Incremental Learner) project is a project that aims at developing theory and methods for AA-learning, following our developmental approach to intelligent robots.

The types of sensor used by SAIL include: visual (video cameras), auditory (microphones), tactile (arm overload sensors, collision detection sensors, etc), and the N-sensor (N stands for "numerical"). An N-sensor accepts text input. Each letter of an alphabet is coded into a feature vector in the sensor output space.

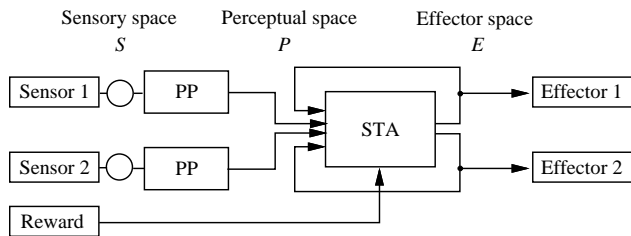


Figure 1: A schematic illustration of the coarse architecture of the presented system SAIL. A circle represents an attention selector. It is also an actuator. PP: preprocessor. STA: spatiotemporal associator.

The types of effector used by SAIL include: attention extractor which extracts a part of the signal vector and applies some weights to the extracted part, pantilt unit (for video cameras), structured speaker (a low-level voice synthesizer), robot arm, vertical slider as the torso, drive base for local motion, and the N-effector, which converts every feature vector into the corresponding text.

SAIL learns continuously for hours and hours daily for its cognitive development and sensory-motion refinement. It starts to learn as soon as the its power is turned on. During learning, human teachers guide the robot to go through training sessions.

The major objective of this article is to discuss the developmental approach. Due to space limitation, we can only present a overview of the SAIL system currently under development.

System Overview

Following the views discussed above, the representation of SAIL must be independent of the content of the information to be processed by the robot. The program-level representation should not be constrained by, or embedded with, handcrafted knowledge-level world models or system behaviors³. It is very difficult to manually build a sufficient set of rules or behaviors that is general and complete enough to handle the challenging recognition tasks we have to deal with. Fig. 1 gives a schematic illustration of the SAIL system architecture. The architecture allows any type of sensors and effectors. The capability to learn from high dimensional sensory space is crucial for such a general-purpose learner. In the following, we briefly discuss SHOSLIF which is used as a key component in the spatiotemporal associator (STA) shown in Fig. 1.

³An example of knowledge-level model is a hand-coded program section that determines something like: “If there are two dark areas of a similar size on a horizontal line, they might be human eyes.” An example of knowledge-level behavior is a hand-coded program section in a system that does something like: “if the distance sensed from the forward sonar is smaller than a certain number, turn away.”

The predecessor: SHOSLIF

SHOSLIF (Weng 1996) (Swets & Weng 1996a) is the predecessor of SAIL. It is a general approximator for supervised or unsupervised learning. However, it does not take time explicitly into account, but SAIL does. SHOSLIF approximates a high dimensional function $f : X \mapsto Y$. It incrementally creates a tree from training samples⁴. It uses the principal component analysis (PCA) and linear discriminant analysis (LDA) to recursively build a different feature space for every internal node of the tree. Each leaf stores the sample x_i it represents. Given any input x , the tree finds the top k matched leaf nodes. For supervised learning, each leaf node stores the desired output. The output $f(x)$ is approximated by a weighted average among the output vectors of top k leaf nodes.

When search proceeds from the root, the SHOSLIF tree rejects, at each tree level, a portion of distant training samples to be examined. It only keeps top- k candidate nodes to be further explored at each tree level. The time complexity for finding top k (constant) matched leaf nodes is $O(\log(n))$ for a tree with n leaf nodes. This extremely low time complexity is important for satisfying the real-time requirement when n is large⁵. SHOSLIF can also learn incrementally by updating the tree. The incremental growing capability of the SHOSLIF is suited for the incremental nature of AA-learning in finding the top k matches.

Real-time continuous learning

Fig. 2 is a flow diagram of the SAIL system. Each level of SAIL is a finite state machine, whose states are created automatically. A self-organization scheme is used to automatically organize, increase, or delete states. Thus, the set of states change according to the experience. The computational structure of the finite state machine can be modeled as a hidden Markov model (HMM). However, it is more general than HMM due to its recurrent nature. The details of the SAIL architecture will appear in another occasion.

The recognition capability of SAIL is expected to help it to realize not only learning-by-trying (reinforcement

⁴A description of the incremental version (Weng & Chen 1996) of SHOSLIF can be download from web <http://www.cps.msu.edu/~weng/research/SHOSLIF-N.html>

⁵To see how low the logarithmic complexity is, suppose a system whose time complexity is $O(\log_b(n))$, regardless of how big the constant coefficient in the time complexity is. If this system can complete a system computation cycle in 100 milliseconds (10 Hz refreshing rate) with 1000 stored prototypes (which is about the speed SHOSLIF has reached), the same program running on another computer whose speed is 4.7 times faster can run at 10 Hz for a network that has stored 10^{14} cases (the absolute number of synapses, or the total numbers stored, in the human brain)! However, the actual capability to handle a large number of prototypes, beyond the currently tried a few thousands, remains to be examined in the project.

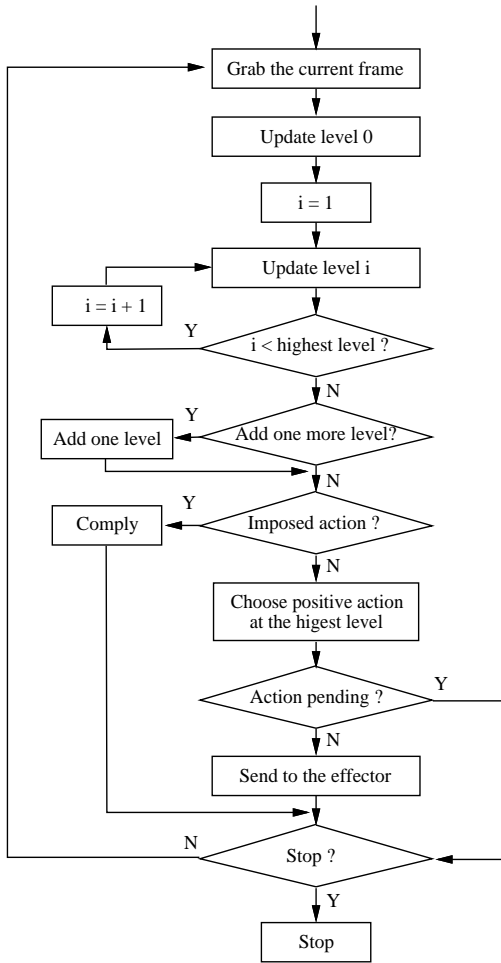


Figure 2: The flow chart of SAIL.

learning), but also learning by being told (supervised learning). It is well known that the latter is much more powerful and efficient than the former. With AA-learning, the channels through which to convey what is to be told include (1) the physical channel and (2) understanding from sensors. The latter capability requires extended, cumulative training through the corresponding sensing modality. For example, a low-pitch tone indicates “move towards me” and a high-pitch tone indicates “move farther away”.

Forgetting

A learner that senses sensory input continuously over a long time must process potentially an infinite amount of data. Due to a finite memory space, the system cannot remember all the sensory-effector associations that it has come across. In fact, it should forget some detail for generalization. Many sensory-effector associations must be forgotten. The utilization of association is indicated by *cooccurrence frequency* and *occurrence internals*.

Let us consider a memory element, a node or a link in STA, which will be used in SAIL for memorizing

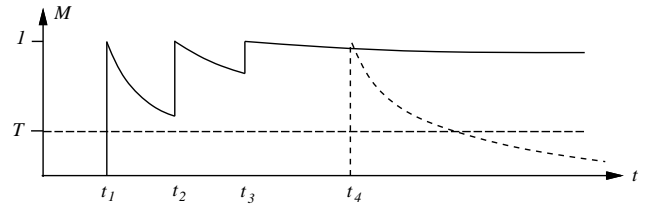


Figure 3: Update of memory trace M through time t . The solid curve represents an element which is visited often enough to be kept. The dashed curve indicates an element that is not visited often enough and thus, it falls below the threshold T before being visited again.

an association it represents. Each element has a memory residual (strength) register whose updating curve is shown in Fig. 3 which resembles what we know about human memory characteristics (Ashcraft 1994) (J. L. Martinez & (eds.) 1991).

Each visit to the same element makes the trace to be reset to 1 and then the curve declines using a next slower speed. For example, we can define a series of memory *fade factors* $\alpha_1 < \alpha_2 < \dots < \alpha_m \approx 1$. α_i is used for an element that has been visited i times. The memory trace r can be updated by $r \leftarrow r\alpha_i^t$ where t is the number of system cycles (refresh) elapsed since the last visit to the element. Thus, we do not need to visit all the elements at every system cycle. When an element is visited, its memory trace is updated first from what remains from the last visit. If the memory trace falls below the designated threshold, it should be deleted and so it is marked as to-be-deleted. If what is deleted is more than a single element (i.e., a subtree), the deleting process will not delete it right away to avoid consuming too much CPU time in a real time process. Instead, it puts the subtree in a garbage buffer which is to be cleaned when the learner is “sleeping.”

The memory fade factors control the relative speed of learning and forgetting so that they can keep in pace while growing the memory with an appropriate speed over time.

Concluding Remarks

A robot that can perform harder, more general tasks must be equipped with more powerful and yet more challenging sensors, such as visual sensors and auditory sensors. However, these sensors bring hard problems from three well-known challenging areas: vision, speech and language. The integration of these three areas is even more difficult, given the known difficulties in each of these areas. However, many hard problems that each area faces cannot be solved within a single area. Integration may find new ways of addressing these problems.

The developmental approach proposed here does not consider integration as a task of integrating a number of developed modules, since this view of the task makes integration extremely difficult to handle. Instead, in-

tegration must be conducted during the entire process of learning. Thus, the architecture for such integrated learning is the crucial point. In other words, (1) the architecture must be designed with integration as one of the primary goals, (2) the integration must start very early (day 1 of the cognitive development), and (3) the integration process must be automated, as is the case with learning itself with the developmental approach. In fact, integration is a part of learning.

The developmental approach proposed in this paper does not treat an intelligent robot as a static system that, once having been built, can be used *as is* in an application environment. Such a robot cannot perform well unless the task is trivial. Learning and adaptation of a robot in the application environment is an integral part of intelligence development. The challenge for continued, cumulative, general purpose learning is how to make a machine that can learn on its own. This is a very important new topic for the robotics and AI community.

The world-knowledge-based approach requires humans to program at the content level of world-knowledge. The behavior-based approach requires humans to program at the content level of behavior. In some sense, both require to manually fabricate a *learned* “brain” for an intelligent robot. On the other extreme, the evolutionary approaches resort to a process of trial-and-selection as the major means to search for an artificial species — intelligent robot. Although various learning methods are more effective than knowledge-based, behavior-based and evolutionary approaches, the special purpose, domain-extensible, unautomated nature of these learning methods determines that the resulting system has only a limited capability in a narrowly defined domain. The proposed developmental approach lets the automatic learning handle the content-level world knowledge and behavior. The automatic learning capability of the robot must be hand-designed, instead of being searched using a genetic algorithm. Such a design task seems more tractable than the tasks of other approaches.

References

Aloimonos, J. 1990. Purposive and qualitative active vision. In *Proc. 10th Int'l Conf. Pattern Recognition*, 346–360.

Ashcraft, M. H. 1994. *Human Memory and Cognition*. New York, NY: Harper Collins College Publishers.

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

Brooks, R. 1991. Intelligence without reason. In *Proc. Int'l Joint Conf. on Artificial Intelligence*, 569–595.

Carey, S. 1985. *Conceptual Change in Childhood*. Cambridge, MA: The MIT Press.

Carey, S. 1990. Cognitive development. In Osherson, D. N., and Smith, E. E., eds., *Thinking*. Cambridge, MA: MIT Press. 147 – 172.

Franklin, S. 1997. *Artificial Minds*. Cambridge, MA: MIT Press.

Gruber, H. E., and Voneche, J. J. 1977. *The essential Piaget*. New York: Basic Books.

Harvey, I. 1992. Evolutionary robotic and SAGA: The case for hill crawling and tournament selection. Technical Report CSRP 222, University of Sussex, Brighton, U.K.

Hwang, W.; Howden, S. J.; and Weng, J. 1996. Performing temporal action with a hand-eye system using the SHOSLIF approach. In *Proc. Int'l Conference on Pattern Recognition*.

J. L. Martinez, J., and (eds.), R. P. K. 1991. *Learning & Memory: A Biological View*. San Diego, CA: Academic Press, 2 edition.

J. R. Deller, J.; Proakis, J. G.; and Hansen, J. H. L. 1993. *Discrete-Time Processing of Speech Signals*. New York, NY: Macmillan.

Laird, J. E.; Yager, E. S.; Hucka, M.; and Tuck, C. M. 1991. Robo-soar: An integration of external interaction, planning, and learning using Soar. *Robotics and Autonomous Systems* 8:113–129.

Lenat, D. B.; Miller, G.; and Yokoi, T. T. 1995. CYC, WordNet, and EDR: Critiques and responses. *Communications of the ACM* 38(11):45–48.

Lenat, D. B. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38(11):33–38.

Mataric, M. J. 1990. A distributed model for mobile robot environment - learning and navigation. Technical Report AI-TR-1228, MIT Artificial Intelligence Laboratory, Cambridge, MA.

Miller, G. A. 1995. Worknet: A lexical database for English. *Communications of the ACM* 38(11):39–41.

Moravec, H. P. 1982. The Stanford Cart and the CMU Rover. *Proceedings of the IEEE* 71(7):872–884.

Newell, A. 1990. *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Rabiner, L. R.; Wilpon, L. G.; and Soong, F. K. 1989. High performance connected digit recognition using hidden Markov models. *IEEE Trans. Acoustics, Speech and Signal Processing* 37:1214–1225.

Schwartz, A. 1993. A reinforcement learning method for maximizing undiscounted rewards. In *Proc. Int'l Joint Conf. on Artificial Intelligence*, 289–305.

Sinha, P., and Poggio, T. 1996. I think i know that face ... *Nature* 384(6608):5.

Swets, D., and Weng, J. 1996a. Discriminant analysis and eigenspace partition tree for face and object recognition from views. In *Proc. Int'l Conference on Automatic Face- and Gesture-Recognition*, 192–197.

Swets, D. L., and Weng, J. 1996b. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18(8):831–836.

Terzopoulos, D.; Tu, X.; and Grzeszczuk, R. 1994. Artificial fishes with autonomous locomotion, perception, behavior and learning in a simulated physical world. In *Artificial Life IV: Proc. Fourth Int'l Workshop on the Synthesis and Simulation of Living Systems*, 17–27. Cambridge, MA: MIT Press.

Turk, M., and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1):71–86.

Watkins, C. 1989. Learning from delayed rewards. Technical report, PhD thesis, King's College, Cambridge, England.

Weng, J., and Chen, S. 1996. Incremental learning for vision-based navigation. In *Proc. Int'l Conference on Pattern Recognition*.

Weng, J. 1996. Cresceptron and SHOSLIF: Toward comprehensive visual learning. In Nayar, S. K., and Poggio, T., eds., *Early Visual Learning*. New York: Oxford University Press.