Grounded Auditory Development by a Developmental Robot [1]

Yilu Zhang and Juyang Weng [2]
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824

Abstract

A developmental robot is one that learns and practices autonomously in the real physical world by interacting with the environment through sensors and effectors, probably under human supervision. The study of developmental robots is motivated by the autonomous developmental process of higher animals and humans from infancy to adulthood. Our goal is to enable a robot to learn autonomously from real-world experiences. This paper presents a case study of a developmental robot developing its auditory related behaviors to follow human trainers' voice command. A learning architecture is proposed to resolve automatic representation generation and selective attention issues. Both simulation results and experiments on real robot are reported to show the effectiveness.

1 Introduction

The development of traditional robots follows a *manual development paradigm*: 1) Given a task, a human engineer analyzes it and translates it into representations and rules that a computer program may work on. 2) The human engineer writes a program that transforms input information into representations and follows human designed rules to control the robot. Segmented off-line sensory data may be used for learning, if the rules are so complex that some parameters cannot be determined by hand. 3) The robot runs the program. There are two major difficulties to build a robot following this traditional paradigm. First, human-collected off-line training data cannot practically cover all the ever changing real-world situations. In other words, the off-line robot learning is never complete. Second, based on pre-processed data, it is hard for a machine to associate sensory information with other physical events, for example, the meaning of a spoken word that can be understood by vision or its own actions, which is called the grounding issue.

Studies of developmental psychology and neuroscience show that a developed adult human brain is an epigenetic product of active, autonomous, extensive interactions with the complex human world [2]. Motivated by this discovery, an *autonomous development paradigm* is discussed in a recent article in *Science [8]*. The new paradigm is as follows: 1) A human designer designs a robot body according to the general ecological condition in which the robot will work (e.g., on-land or underwater); 2) A human programmer designs a task-nonspecific developmental program for the robot; 3) The robot starts to run the developmental program and develops its mental skills through real-time, online interactions with the environment which includes humans (mind development). A robot built through the autonomous development paradigm is called a *developmental robot*. Early examples of such developmental robots include SAIL (short for Self-organizing, Autonomous, Incremental Learner) robot [7] at Michigan State University and Darwin V robot [1] at The Neurosciences Institute, San Diego.

While the developmental program is the central part of a developmental robot, the step of "mind development" is the key in this new paradigm. In this step, a robot generates its representations of the world autonomously according to the task it encounters. What the robot behaves is also determined in this step by the experiences accumulated in the real world. In contrast to the traditional paradigm, learning happens *after* the system is released for final usage instead of in the process of building the system. Intelligence is a capability to solve multiple tasks. With the factor of learning after "born" we may reach a possibility of a task-nonspecific solution.

[1] The work is supported in part by National Science Foundation under grant No. IIS 9815191, and DARPA ETO under contract No. DAAN02-98-C-4025, DARPA ITO under grant No. DABT63-99-1-0014.

[2] The authors would like to thank Wey S. Hwang for his major contribution to an earlier version of the IHDR program.

This paper presents a case study of a developmental robot developing its auditory related behaviors to follow human trainers' voice command.

2 Challenging Issues

Clearly, such an autonomous development requirement compounds many difficulties.

Automatic generation of internal representation. For a traditional learning robot, representations of a task are typically decided manually. A human engineer takes the advantage of knowing the task and the environment by programming various constraint into the robot program. For the robot system we are interested in, the task-specific information is not available until the system is built and released to the users, when the internal-data-level intervention by human engineers is no longer possible. Interactions between the robot and its environment are only through sensors and effectors. Therefore, a developmental robot must collect data, derive useful features, generate internal representation, and acquire behaviors automatically from raw sensory signals.

Unsegmented sensory stream. The environment is sensed by a robot as continuous and endless data streams, e.g., the audio stream from a microphone, visual streams from cameras, tactile streams from touch sensors, etc. To handle such sequential information, one important issue is to decide appropriate context. For example, meaningful speech units (phonemes, words, phrases, and sentences) have varying temporal length. Without the help of a human designer to edit or transcribe sensory input, a robot must automatically associate an appropriate context (state) with a desired output. This association is established according to certain feedback the robot receives from the environment, such as the reward supplied by humans (encouraging or discouraging signals) or nonhuman environment (collisions). However, a reward is typically delayed and is inconsistent in the time at which the reward is delivered.

Learning internal behaviors. Autonomous learning is not effective without developing internal behaviors. Internal behaviors are perception invoked actions that are applied to internal effectors, such as attention effectors, action release effectors, etc. For example, closely related to the issue of unsegmented sensory input is the need for a selective attention behavior, which allows a robot to focus on the part of the input data that is critical in current situations. Another example of internal behaviors is the manipulation of internal states. Since internal behaviors are not accessible from the outside world, it is vary challenging to develop such behaviors through external interactions via robot sensors.

3 System Architecture And Learning Strategy

Shown in Fig. 1 is the basic architecture that has been implemented for the work presented here.

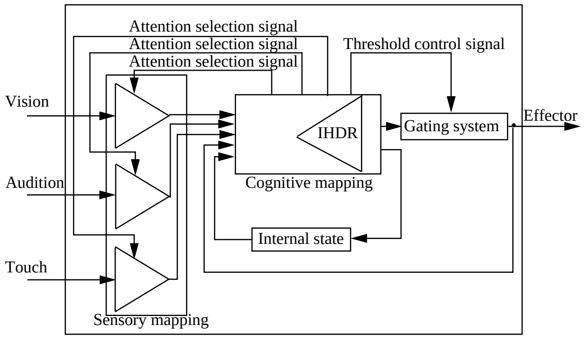

Figure 1: Basic architecture of SAIL experiment.

A sensory input first enters a module called sensory mapping. A more detailed structure of the sensory mapping module is shown in Fig. 2. This module divides the temporal and spatial input space into overlapped regions of different sizes, named receptive fields. Each of these regions corresponds to a processing unit which is called neuron here for convenience. These neurons are organized in a hierarchical manner so that neurons at lower layers have smaller receptive fields. The outputs from lower-layer neurons are used as input to neurons at next higher layer. Thus, neurons at higher layers have a larger receptive field. Each neurons performs principal component analysis (PCA) which maps the input high-dimensional data into output space with a lower dimension while keeping as much data variance as possible. The neurons at the lowest layer also do sensor-specific preprocessing, such as cepstral analysis for auditory modality. The vector outputs from these neurons are fed into the next module, called cognitive mapping, after being selected by the control signals from the cognitive mapping. This selection is a part of internal behaviors that enables the selective attention mechanism.

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

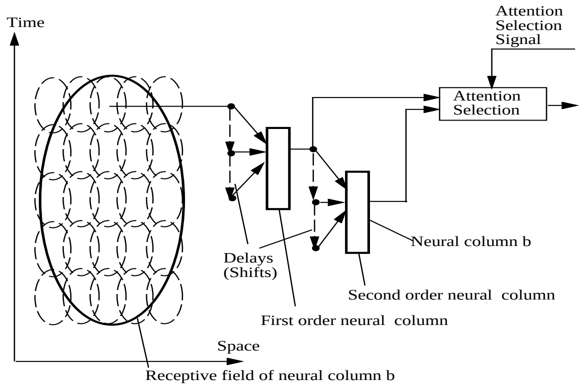
Time
Attention
Selection
Signal
Attention
Selection
Delays
(Shifts)
Neural column b
Second order neural column
First order neural column
Space
Receptive field of neural column b

Figure 2: Receptive field of sensory mapping

for the sensory mapping module, the effector that manipulates the internal states and the threshold control signals to the gating system in Fig. 1.

Mathematically, the cognitive mapping computes the mapping $g : S \times X \rightarrow S \times A$, where S is the state (context) space, X is the sensory space, and A is the action space. In other words, g accepts the current state $s(t)$ and the current sensory input $x(t)$ to generate the new state $s(t+1)$ and the corresponding output $a(t+1)$. The cognitive mapping is realized by the Incremental Hierarchical Discriminant Regression (IHDR) tree, which finds the best features that are most relevant to output and disregard irrelevant ones. IHDR uses a tree structure to find the best matching input cluster in a fast logarithmic time. Compared to other methods, such as artificial neural network, linear discriminant analysis, and principal component analysis, IHDR has advantages in handling high-dimensional input, doing discriminant feature selection, and learning from one instance. A more detailed explanation is beyond the scope. The reader is referred to [4] [9]. The internal state effectively covers a few frames of sensory inputs. While the state may be manipulated by internal behaviors, implemented here is simply the concatenation of the frames.

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

Although we report here only the work related to auditory and tactile inputs, the architecture proposed can deal with different modalities.

3.1 Learning Strategy

Supervised learning enables mapping from input space to output space using a set of given input and output pairs. This learning mode is efficient when the input-output pairs are continuously available, e.g., in the case of vehicle control when human teacher holds the steering wheel during training.

There are situations when supervised learning is not practical. For example, the robot needs internal behaviors such as selecting receptive fields and manipulating internal states. The trainer does not have access to these internal actions and may only encourage or discourage them through a reinforcement learning procedure. With reinforcement learning, the system autonomously generate actions and the human teacher needs only give appropriate rewards after the actions.

Q-learning is one of the most popular reinforcement learning algorithms [5] [6]. In Q-learning, each state maintains a set of action values, called Q-values. The action with the largest value will be selected as the system output. The Q-learning algorithm is as follows,

$$\begin{aligned} Q(s(t-1),a(t)) \leftarrow & (1-\alpha)Q(s(t-1),a(t)) \\ & + \alpha(r(t) + \gamma \max_{a'} Q(s(t),a')); \quad (1) \end{aligned}$$

where a and a' are actions associated with a state, $s(t-1)$ is the state at time $t-1$, $s(t)$ is the state the system lands on after executing action $a(t)$, $r(t)$ is the reward received from the environment, α and γ are learning rates. The algorithm shows that Q-values are updated according to the immediate reward $r(t)$ and the value of the next state, which allows delayed reward to be back-propagated in time during learning. This way, the system can predict the reward correctly when similar context (state) occurs next time.

We have adapted Q-learning to integrate supervised learning and reinforcement learning strategies into a unified mode. In our integrated learning mode, if no action is imposed by the environment, we follow the Q-learning algorithm to update the action values. If an action is imposed by the environment, we increase the value of the corresponding action with a fixed amount. Typically, we make this amount large so as to reflect the fact that the system should follow the instructions of the trainer.

At each time instance, the system executes following learning procedure, where $f(t)$ is the new audio frame, $a(t)$ is the current action vector and $r(t)$ is the current reward.

```
Learn($f(t)$,$a(t)$,$r(t)$) {
```

Table 1: Results on digit recognition

External Behavior	1	2	3	4	5	6	7	8	9	10	Average
Correct Rate(%)	98.4	95.2	93.7	96.8	95.2	93.7	96.8	92.1	93.7	93.7	94.9
Incorrect Rate(%)	1.6	3.2	3.2	3.2	4.8	3.2	3.2	3.2	3.2	3.2	3.2
Rejection Rate(%)	0	1.6	3.2	0	0	3.2	0	4.8	3.2	3.2	1.9

```
    1. Grab the new audio frame $f(t)$;
    2. Update each neural column of the sensory mapping module by doing PCA incrementally;
    3. Concatenate the outputs from the neural columns in the sensory mapping module and the current external action vector, $a(t)$ into a single vector, $x(t)$;
    4. Replace the oldest part of the internal state $s(t-1)$ with $x(t)$ to get a vector $S(t)$;
    5. Query the IHDR tree and get a prototype, $s(t)$ that is closest to $S(t)$ using the fast tree search;
    6. If $S(t)$ is significantly different from $s(t)$, it is considered as a new sample and we update IHDR tree by saving $S(t)$. Otherwise, $S(t)$ updates $s(t)$ through incremental averaging;
    7. If $a(t)$ is given, increment the value of the action that is most similar to $a(t)$ and return.
    8. Otherwise, update $Q$-values of $s(t-1)$ with Eq. 1 and return $\arg\max_{a'} Q(s(t), a')$ as the new action at this time instance.
}
```

4 Simulation Results

In the experiment presented here, we trained the robot to act correctly according to real-time online auditory inputs.

The effector of the system is represented by a 10-D action vector. 10 desired behaviors are defined, each for one of the 10 digits ("one" to "ten"). Behaviors are identified by the component of the action vector with the maximum value. For example, if the first component of the action vector has the maximum value, it is identified as action 1.

The sensory mapping module has 4 layers. The neural column in each layer takes as the input the outputs from 10 neural columns of the next lower layer. 13-order Mel-frequency Cepstral Coefficients (MFCCs) [3] are computed before the data reach the sensory mapping module. The auditory data are fed into the system continuously during both training and test sessions.

The training procedure consists of two phases, supervised learning followed by reinforcement learning.

In supervised learning, the imposed action is given to the system by the end of each utterance. In reinforcement learning, each reward is decided as follows. If the system makes an action at the end of an utterance, and the action is correct, the robot receives a reward 1. If the action is wrong, the system gets a reward -1. If no action is made within the time window, the system gets a reward -1. In all other cases including silence period, the system gets reward -0.001.

The auditory data were collected as follows. 63 persons with a variety of nationalities, including American, Chinese, French, India, Malaysian and Spanish, and ages, from 18 to 50, participated in our speech data collection. Each person made 5 utterances for each of the 10 digits. There is a silence of a length of about 0.5s between two consecutive utterances. This way, we got an speech data set with totally 3150 isolated utterances.

The performance was evaluated as follows. Within a short period before and after the end of an utterance, if there is one incorrect action or if the system does not take any action, we marked it as an error. If the system reacted correctly once or more than once within that time window, we marked it as correct. The test was done using 5-fold leave-one-out cross-validation. The results are summarized in Table 1. The robot learned reliable responses through this challenging online interactive learning mode.

5 Experiment On Selective Attention Learning

To specifically test the system's attention selection capability, we conducted another experiment using purely reinforcement learning.

The system architecture here is similar to the one in last section except that it has only two layers in sensory mapping module and there are internal actions to select the output from one of these two layers. A higher layer output corresponds to input of a longer temporal sensory sequence. This is an example to describe how the robot learned to select an appropriate context

Table 2: Attention selection behavior per external behavior

External Behaviors	1	2	3	4	5	6	7	8	9	10
1st layer	0	1.00	0.73	0.83	0.66	0.74	0	0.63	0.25	0.40
2nd layer	1.00	0	0.27	0.17	0.33	0.26	1.00	0.38	0.75	0.60

from multiple contexts of different temporal lengths. We randomly chose 5 persons to train the system. Its overall performance improves consistently as the training iterations go on (Fig. 3).

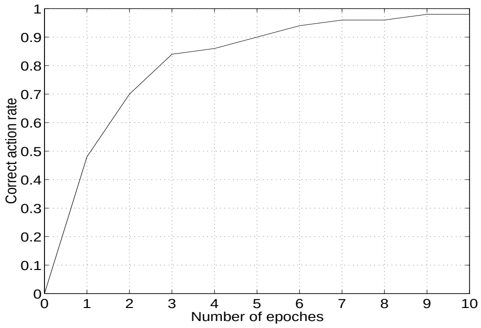

Figure 3: The overall performance vs. training epoches

After training for 10 epoches, we examined the system's choices on attention selection when it makes external behaviors. As shown in Table 2, the first line includes the external behaviors represented by their corresponding digit utterances. The second and third lines show the probability that the system selects 1st or 2nd layer when firing external behaviors.

To explain how the robot learned to chose context of correct length, let us examine the cases of 1 and 7. The system exclusively chooses the second layer when making decision for them. Why? The tail parts of the utterances of "one" and "seven" are very similar! In these cases, the first layer does not cover enough context for a correct discrimination. The second layer is the right source of context. Other sounds vary in context depending on speaker variation. In other words, the attention is not programmed in, but learned according to each word, each speaker and even each utterance.

6 Experiment On SAIL Robot

The next experiment is to control the SAIL robot (see Fig. 4) by teaching it voice commands. SAIL is a human-size robot custom-made at Michigan State University. SAIL's "neck" can turn. Each of its two "eyes" is controlled by a fast pan-tilt head. Its torso has 4 pressure sensors to sense push actions and force. It has 28 touch sensors on its arm, neck, head, and bumper to allow human to teach how to act by direct touch. Its drive-base is adapted from a wheel-chair and thus SAIL can operate both indoor and outdoor. Its main computer is a high-end dual-processor dual-bus PC workstation with 512 MB RAM and an internal 27 GB three-drive disk array for real-time sensory information processing, real-time memory recall and update as well as real-time effector controls.

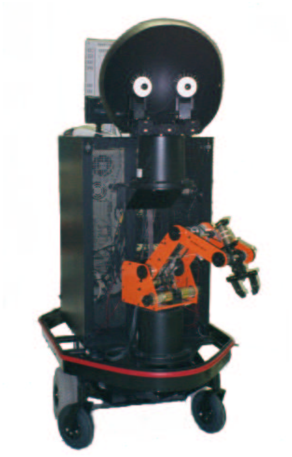

Figure 4: SAIL robot at Michigan State University.

To reduce the learning period, we only used supervised learning mode in this experiment. The learning process is conducted online in real-time through physical interactions between a trainer and the SAIL robot. The trainer speaks to the robot with a spoken command C and then executes any desired action A by corresponding pushing the pressure sensor or the touch sensor. The verbal commands include "go left", "go right", "forward", "backward", "freeze", "arm left", "arm right", "arm up", "arm down", "hand open", "hand close", "see left", "see right", "see up", "see down". After training for 15 minutes, the SAIL robot could follow commands with about 90% correct rate.

To further test the capability of dealing with speaker variation, we have conducted a corresponding multi-trainer experiment. Eleven persons participated in training. They spoke each of 15 commands for 5 times which resulted in 825 utterances. These data (4 out of 5 utterances of each commands) were fed into the SAIL robot off-line appended with appropriated actions at the end of each utterance. The SAIL robot with partially trained "brain" started to run in real-

Table 3: Performance of the SAIL robot when following the 12th trainer's command

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

Table 4: Performance of the SAIL robot on off-line test data

Commands	Go left	Go right	Forward	Backward	Freeze
Correct rate(%)	94.5	89.9	92.7	100.0	100.0
Commands	Arm left	Arm right	Arm up	Arm down	Hand open
Correct rate(%)	100.0	90.9	96.3	92.7	89.9
Commands	Hand close	See left	See right	See up	See down
Correct rate(%)	89.9	90.9	92.7	100.0	100.0

time. Then, a trainer, being the 12th trainer, taught the SAIL robot through physical interactions 4 times for each command. In this way, we simulated the situation that a partially developed SAIL robot continuously developed its audio-driven behaviors online.

After training, the 12th trainer tested the SAIL robot by guiding it through the second floor of Engineering Building. As SAIL did not have perfect heading alignment, the human trainer used verbal commands to adjust robot heading during turns and straight navigation. During the navigation, the arm and eye commands are issued 10 times each at different locations. The performance is summarized in Table 3. The performance for those trainers, who did not have chance to test the SAIL robot in real-time, was evaluated using the left-out utterances to test it off-line. The performance is summarized in Table 4.

7 Conclusions

We have demonstrated that it is possible for a machine to develop its behaviors through challenging real-time interactions with the environment. This progress has realized the first successful online robot learning of verbal commands. Our online learning is fundamentally different from other online verbal learning devices in that out robot does not “know” what will possible be spoken when it starts to run but the latter does (and thus a correct word model is waiting). Technically, IHDR played a great role in this success. It automatically derives features and thus automatically generates internal representations. The coarse-to-fine memory self-organization of IHDR ensures a logarithmic time complexity. The robot develops its intramodal attention mechanism by learning internal behaviors to select proper temporal context. The result reported here is a step toward a robot that can learn automatically, like an animal or human.

References

[1] N. Almassy, G.M. Edelman, and O. Sporns. Behavioral constraints in the development of neuronal properties: a cortical model embedded in a real-word device. *Cerebral Cortex*, 8:346–361, June 1998.

[2] M. Cole and S. R. Cole. *The Development of Children*. Freeman, New York, third edition, 1996.

[3] J. R. Deller, Jone G. Proakis, and John H. L. Hansen. *Discrete-Time Processing of Speech Signals*. Macmillan, New York, NY, 1993.

[4] W. Hwang and J. Weng. Hierachical discriminant regression. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11), 2000.

[5] R.S. Sutton and A.G. Barto. *Reinforcement Learning – An Introduction*. The MIT Press, Cambridge, MA, 1998.

[6] C. J. Watkins. Q-learning. *Machine Learning*, 8:279–292, 1992.

[7] J. Weng. The living machine initiative. Technical Report CPS 96-60, Department of Computer Science, Michigan State University, East Lansing, MI, Dec. 1996.

[8] J. Weng and et. al. Autonomous mental development by robots and animals. *Science*, 291:599–600, January 26 2000.

[9] J. Weng and W. Hwang. An incremental learning algorithm with automatically derived discriminating features. In *Proc. Fourth Asian Conference on Computer Vision*, pages 426–431, Taipei, Taiwan, January 8-9, 2000.