ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 117-122

# Incremental Category Learning in a Real World Artifact Using Growing Dynamic Cell Structures

Christian Scheier

AI Lab, Computer Science Department, University of Zurich
Winterthurerstrasse 190, 8057 Zurich, Switzerland
E-mail: scheier@ifi.unizh.ch

**Abstract.** A biological or artifical agent operating in the real world must have means to categorize its environment. Since the real world is dynamic the problem arises how an agent can adaptively categorize non-stationary input data. In this paper we address this stability-flexibility trade-off using a mobile robot that has to solve a collecting task in a changing environment. We demonstrate that by using a Growing Dynamical Cell Structures algorithm the robot can incrementally learn categories.

## 1. Introduction

If a biological or artificial agent is to function in the real world it must have means to categorize its environment. In previous work we have developed a new approach to address the problem of adaptive categorization in mobile robots ([2],[4]). The main idea is to reduce the many degrees of freedom of the input space by exploiting correlations through time-linked independent samples of sensory stimuli and kinesthetic signals produced by self-motion. Thus, categorization is not seen as an isolated perceptual (sub-)system but rather as a *sensory-motor coordination.* In these experiments a self-organizing map (SOM) was used for the category learning. Here we extend this work by using a Growing Dynamical Cell Structures algorithm (GDCS) ([1]) instead of a SOM. The main goal is to address the problem of categorization in a dynamic environment. We use a GDCS network because it has been shown to have the same topology preserving properties as standard SOMs ([3]) but in addition allows for incremental learning of inputs with changing probability densities. This is a crucial feature with respect to categorization in the real world: since the real world is dynamic an agent has to be able to adapt to these changes without forgetting what it has learned previously. In the experiments presented in this paper a mobile robot has to learn to interact with different types of objects (see figure 1). Mixtures of new and already learned objects are successively presented to the robot. The robot has to learn the new objects while not forgetting the ones already encountered, i.e. it has to solve a stability-flexibility trade-off.

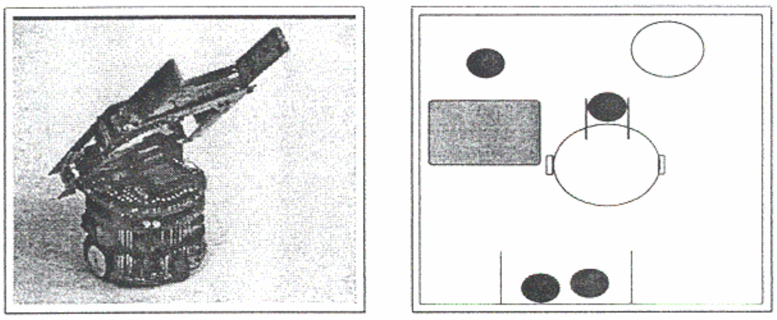

Figure 1: The robot used in the experiments is a $Khepera^{TM}$. It is equipped with eight IR sensors, six in the front and two in the back. The arm is moved by a DC motor coupled with a position sensor. Objects inside the gripper can be detected by an optical barrier that is mounted on the gripper. Two conductive surfaces on the gripper allow to measure the electrical conductivity of an object. The robot's task is to collect small non-conductive objects (small black circles in the figure), to push intermediate sized conductive objects (indicated as a white circle) and to avoid large objects (indicated as a shaded rectangle)

## 2. Control Architecture

The control architecture is based on an Extended Braitenberg Architecture ([4]). It is illustrated in the left part of figure 2. There are a number of *behavioral neural networks* (BNN) functioning in parallel. Each BNN receives weighted input from sensors and effectors (proprioception) and from other networks. They all write simultaneously onto the effector variables where they are summed by a particular summation scheme. The resulting values of these effector variables determine the behavior of the agent. The behavior of the robot is as follows (for a detailed formal description of the BNN's used in this paper see ([4]). If there is no activation in the other BNN's the robot moves forward until it encounters an object. When the robot is close to an object it tries to avoid a collision by turning away from it. Whenever the right- or left-most IR sensor is activated, the *turn towards object* network tries to maintain this condition. As a result of the simultaneous activation of this network and the *avoid* network the robot will start exploring objects it encounters head on by circling around them. Each time the robot explores an object it senses to object after a predefined exploration time by lowering the gripper. The units in the *grasp* and the *push* network respond to activation of the resistivity sensor. If the object is conductive the resistiviy sensor is on otherwise it remains off. The units in the *turn-away* network receive input from the position sensor. For large pegs the gripper cannot be lowered which results in large values of the position sensor. Another source of input to the *grasp*, *push* and *turn-away* networks is the GDCS (see figure 2). The GDCS network receives input from the IR sensors and from a node which encodes angular velocities. This implements the notion of sensory-motor coordination: the GDCS network learns a topographical mapping of sensory

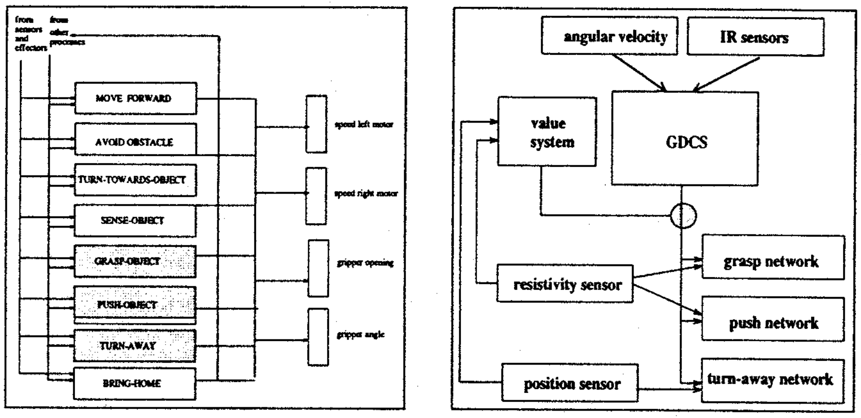

Figure 2: The control architecture of the robot. The behavioral networks are depicted on the left. The shaded networks are connected to the GDCS network. The details of these connections is shown on the right.

(IR sensors) and kinesthetic (angular velocities) states whenever the robot is exploring an object. The GDCS network is connected via modifiable weights to the three behavioral networks which implement grasping, pushing and turning away behavior, respectively. Changes in the weight strengths can be modulated by a value signal. The main idea behind these adaptive connections is that the agent should learn to associate its sensory-motor mappings or categories with these behaviors. The value system receives input from the *turn-towards-object* network, the position as well as the resistivity sensor. The value system essentially provides the agent' s own reinforcement for the actions taken. In sum, if the object is non-conductive the robot grasps it, if not it pushes it and if the resistivity cannot be read (indicated by large values in the position sensor) it turns away from the object. These are reflex actions. Learning occurs whenever the robot makes an action, and the weights between the GDCS and the BNNs connected to it evolve. As a result of this learning process an association between the sensory-motor categories and grasping, pushing or avoiding an object is acquired. After learning the robot will execute these actions after a short exploration time without sensing the object.

## 3. Learning Rules

One important characteristic of the GDCS algorithm is that the lateral connections of neurons are modifiable. Lateral connections are non-symmetric and updated according to a *competitive Hebb rule*:

$$C_{ij} = \begin{cases} 1 & : \; i = bmn \wedge j = second \\ 0 & : \; i = bmn \wedge (\alpha \cdot C_{ij} < \theta \wedge C_{ji} < \theta) \\ \alpha \cdot C_{ij} & : \; i = bmn \wedge (\alpha \cdot C_{ij} \geq \theta \vee C_{ji} \geq \theta \\ C_{ij} & : \; otherwise \end{cases} \tag{1}$$

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 117-122

where $C_{ij}$ is the connection between neuron $i$ and neuron $j$, $bmn$ is the current best matching neuron. The centers $c$ are adapted using an *error modulated Kohonen rule* such that the distribution of units converges to a uniform distribution of local approximation errors:

$$\begin{aligned} \Delta c_{bmn} &= f(\frac{\bar{r}}{r_{bmn}}) \cdot \varepsilon_{bmn} \cdot (x - c_{bmn}), \\ \forall n \in Nh(bmn) : \Delta c_n &= f(\frac{r_n}{r_{bmn}}) \cdot \varepsilon_{Nh} \cdot (x - c_n) \end{aligned} \tag{2}$$

where $Nh(bmn)$ are the neighboring units of the bmn, $r_{bmn}$ is a moving average of error signals of the $bmn$, $x$ is the input, $\bar{r}$ is the mean error of $Nh(bmn)$ and $f(\cdot)$ is a monotonically decreasing function with $c(0) = 1$ and $c(1) = 0$. A *distance driven insertion* strategy inserts new neurons at the position of the input if the distance of the input to the center of the $bmn$ exceeds a certain threshold. Distance driven insertion is crucial for the incremental category learning experiments presented below because it inserts new neurons at the input positions of new categories. An *error driven insertion* strategy inserts new neurons between the neurons with the largest resource values. This leads to a fine grained approximation of the input data. The updating of the weights between the GDCS network and the behavioral networks is as follows:

$$\Delta w_{ij} = v(t)(\eta a_i(t)a_j(t) - \epsilon((a_i(t) + a_j(t))w_j(t)) \tag{4}$$

where $w_{ij}$ represents the strength of the connection between the unit $j$ of the GDCS and the $i$-th BNN, $v(t)$ is the value signal (activity of the value sytem), $a_j$ is the activation of neruon $j$ in the GDCS, $a_i$ is the activation of the $i$-th BNN, and $\eta, \epsilon$ are the learning rate and decay parameters, respectively. Thus, weights are decreased if there is either no pre- or no postsynaptic activity. The main advantage this *bidirectional active forgetting* is that erroneous associations between the GDCS network and the behavioral networks that have been built up in the beginning of the learning process - where the sensory-motor mappings have not yet been stabilized - will be significantly decreased as soon as a stable mapping is acquired.

## 4. Results

There were 3 experimental stages. In a first phase (PI) only small (non-conductive) objects were presented to the robot. Then a mixture of small and (conductive) objects of intermediate size were presented (PII). Finally, a mixture of small, intermediate sized, and large objects were presented (PIII). In order to solve the stability-flexibility trade-off the robot had to (a) adapt to new objects and (b) not forget already learned objects. The parameter values of the GDCS were as in ([1]) and the ones of the GDCS-BNN connections as in ([2]). The initial size of the GDCS was set to 2 neurons in all trials. The training of the GDCS was based on a moving window of 60 samples. Samples were 9-dimensional vectors

| Phase | Reflex Triggered | Category Triggered |
|---|---|---|
| PI | S: 5.1±0.5 | S: 44.6±3.4 |
| PII | S: 1.6±0.2 I: 5.6±0.6 | S: 48.6±1 I: 39.8±4.2 |
| PIII | S: 1.9±0.3 I: 2.3±0.5 L: 6.2±1.1 | S: 48.1±1.2 I: 47.6±1.3 L: 42.8±4.5 |

Table 1: Averaged (±sdev.) sums over 5 trials and 50 pegs of each type. PI-PII: Experimental phases. S - small objects, I - intermediate sized objects, L - large objects

consisting of the current readings from the eight IR sensors and the moving average of absolute angular velocitiy values. The overall performance is shown in table 1. The main result to be taken from table 1 is that (a) a stable mapping in the GDCS which allows the GDCS-BNN weights to evolve appropriately is acquired very quickly (after less than 7 objects), (b) new categories are learned incrementally when they are encountered in a new phase, and (c) categories that have been learned are not forgotten in subsequent phases. The GDCS dynamics leading to this behavior are depicted in figure 3.

## 5. Conclusion and Future Work

In this paper we have demonstrated that incremental category learning in a mobile robot operating in a changing environment can be achieved by using a Growing Dynamical Cell Structures algorithm. Future work will include the investigation of forgetting dynamics in cases where categories are removed completely from the environment. Moreover, the GDCS algorithm will be refined according to ideas used in the temporal SOM.

## References

[1] J. Bruske I. Ahrns and G. Sommer. On-line learning with dynamic cell structures. In *Proceedings of the International Conference in Artificial Neural Networks ICANN'95*, pages 141–146, Paris, October 1995.

[2] D. Lambrinos, C. Scheier, and R. Pfeifer. Unsupervised classification of sensory-motor states in a real world artifact using a temporal kohonen map. In *Proceedings of the International Conference in Artificial Neural Networks ICANN'95*, pages 467–472, Paris, October 1995.

[3] T. Martinez. Toplogy representing networks. *Neural Networks*, 7:505–522, 1994.

[4] C. Scheier and R. Pfeifer. Classification as sensory-motor coordination: a case study on autonomous agents. In *Proceedings of the Third European Conference on Artificial Life ECAL95*, pages 656–667, Granada, Spain, June 1995.

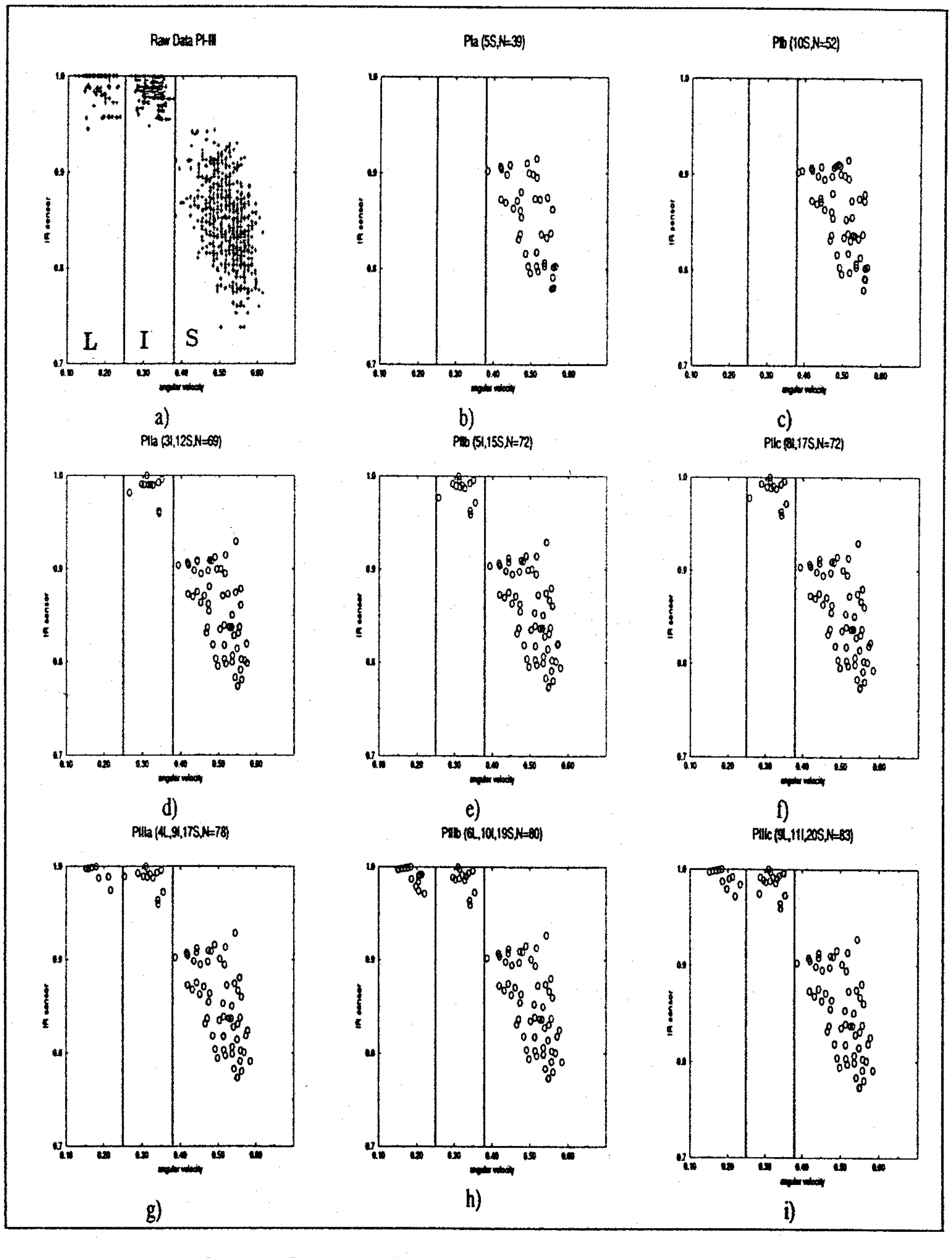

**Figure 3: Raw data and GDCS dynamics. Only non-zero entries in the sample vectors and the GDCS centers were used for visualization. (a) raw data. (b)-(i) snapshots of typical GDCS dynamics. Titles indicate phase, type and number of objects, and number of neurons. For example, in (b) 5 small pegs were encountered in phase PI and 39 neurons were evolved.**