

Research Article

Haptic Perception with Self-Organizing ANNs and an Anthropomorphic Robot Hand

Magnus Johnsson and Christian Balkenius

Lund University Cognitive Science, Kungshuset, Lundagård, 222 22 LUND, Sweden

Correspondence should be addressed to Magnus Johnsson, magnus.johnsson@lucs.lu.se

Received 18 August 2009; Accepted 30 December 2009

Academic Editor: Noriyasu Homma

Copyright © 2010 M. Johnsson and C. Balkenius. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We have implemented and compared four biologically motivated self-organizing haptic systems based on proprioception. All systems employ a 12-d.o.f. anthropomorphic robot hand, the LUCS Haptic Hand 3. The four systems differ in the kind of self-organizing neural network used for clustering. For the mapping of the explored objects, one system uses a Self-Organizing Map (SOM), one uses a Growing Cell Structure (GCS), one uses a Growing Cell Structure with Deletion of Neurons (GCS-DN), and one uses a Growing Grid (GG). The systems were trained and tested with 10 different objects of different sizes from two different shape categories. The generalization abilities of the systems were tested with 6 new objects. The systems showed good performance with the objects from the training set as well as in the generalization experiments. Thus the systems could discriminate individual objects, and they clustered the activities into small cylinders, large cylinders, small blocks, and large blocks. Moreover, the self-organizing ANNs were also organized according to size. The GCS-DN system also evolved disconnected networks representing the different clusters in the input space (small cylinders, large cylinders, small blocks, large blocks), and the generalization samples activated neurons in a proper subnetwork in all but one case.

1. Introduction

Haptic perception, that is, active tactile perception, is of utmost importance in the field of robotics since a well-performing robot must be able to interact with objects in its environments. However, haptic perception is also important in supporting and sometimes also in substituting the visual modality during the recognition of objects. Like humans, robots should be able to perceive shape and size as well as to discriminate between individual objects by haptic exploration.

The modelling of haptic perception as well as the implementation of haptic perception in robots have been neglected areas of research. Robot hand research has mainly focused on grasping and object manipulation [1–4], and many models of hand control have focused on the motor aspect rather than on haptic perception [5, 6], although there are some exceptions [7–17].

Previously we have designed and implemented haptic size perception systems [18–21], haptic shape perception

systems [22–25], and haptic texture/hardness perception systems [26, 27].

The haptic size perception systems used a simple three-fingered robot hand, the LUCS Haptic Hand I, with the thumb as the only movable part. The LUCS Haptic Hand I was equipped with 9 piezo-electric tactile sensors. This system used Self-Organizing Maps, SOMs, [28] and a neural network with leaky integrators and it successfully learned to categorize a test set of spheres and cubes according to size.

The haptic shape perception systems used a three-fingered 8 d.o.f robot hand, the LUCS Haptic Hand II, equipped with a wrist for horizontal rotation and a mechanism for vertical repositioning. This robot hand was equipped with 45 piezo-electric tactile sensors. This system used active explorations of the objects by several grasps with the robot hand to gather tactile information. The LUCS Haptic Hand II was not equipped with any proprioceptive sensors, that is, sensors that register joint angles, but the system used the positioning commands to the actuators as a substitute. Depending on the version of the system, either

tensor product (outer product) operations or a novel neural network, the Tensor Multiple Peak SOM, T-MPSOM [23–25], was used to code the tactile information in a useful way while a SOM was used for the categorization. The system successfully learned to discriminate between different shapes as well as between different objects within a shape category when tested with a set of spheres, blocks, or cylinders.

The haptic texture/hardness perception systems employed a microphone-based texture sensor and a hardness sensor that measures the displacement of a stick pressed at the object with a constant force. With these sensors, we implemented systems that automatically evolved monomodal as well as bimodal representations of texture and hardness [26], and also a system that evolved monomodal representations of texture and hardness while at the same time learning to associate these representations. The latter was done by using a variant of the SOM called Associative Self-Organizing Map (A-SOM) [27].

This paper explores a somewhat different approach to shape and size perception which is based solely on proprioception. Using the position of each joint as the only input, we have designed an anthropomorphic robot hand and self-organizing systems that can discriminate objects and categorize them according to shape and size [29–31].

When designing a neural network based on self-organizing perception system a natural question comes up, namely, what kinds of neural network architectures are most suitable to use. A common choice is the self-organizing map (SOM) that we have used in previous work. This is often a very good choice but it suffers from some limitations, for example, the topological structure is fixed and the number of neurons in the neural network has to be preset by the system designer. Other limitations are that parameters like the learning rate, the initial neighbourhood size, and the decreasing rate of the neighbourhood size also have to be set manually by the designer.

To address these limitations we have, in addition to a SOM-based system, explored and compared three other haptic systems based on the same robotic hand. These systems are based on alternative neural network architectures that avoid some or all of the limitations with a SOM-based system. The four systems differ in one respect, namely, in the kind of self-organizing neural network employed to cluster the input. The first system uses the SOM, the second uses the Growing Cell Structures (GCS), the third uses the Growing Cell Structures with Deletion of Neurons GCS-DN [32, 33], and the fourth uses the Growing Grid (GG) [34].

2. LUCS Haptic Hand III

The LUCS Haptic Hand III is a five-fingered 12-d.o.f anthropomorphic robot hand equipped with 11 proprioceptive sensors (Figure 1). The robot hand has a thumb consisting of two phalanges, whereas the other fingers have three phalanges. The thumb can be separately flexed/extended in both the proximal and the distal joints and adducted/abducted. The other fingers can be separately flexed/extended in their proximal joints, whereas the middle and the distal joints are

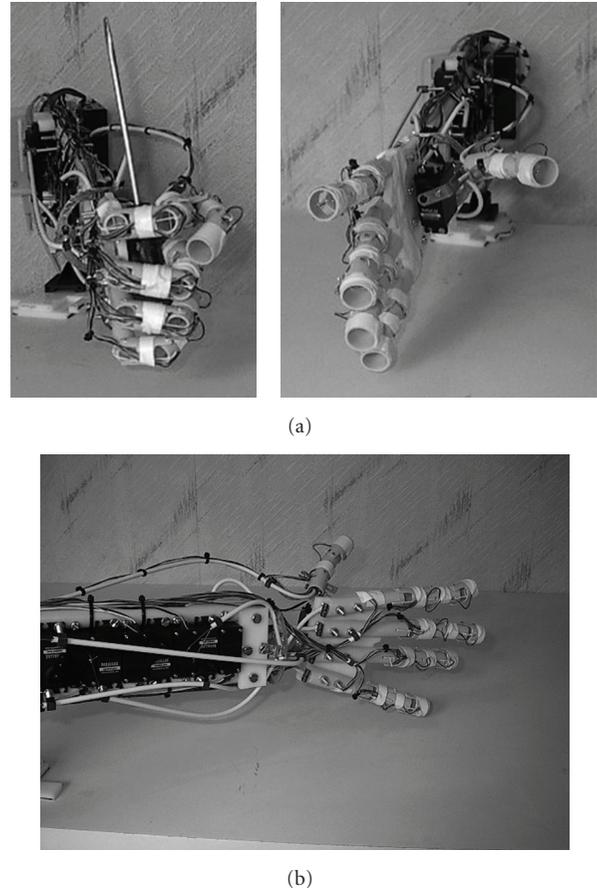


FIGURE 1: The LUCS Haptic Hand III while holding a screw driver, in open position seen in a front view and in a side view. Some of the actuators in the forearm can also be seen in the side view. The 12-d.o.f robot hand has five fingers, is of the same size as a human hand, and all its parts have approximately the same proportions as their counterparts in a human hand. Each finger can be separately flexed/extended in the proximal joint, whereas the medial and distal joints are flexed/extended together as real human fingers. As a human hand, the thumb has only a proximal and a distal phalanx. These can also be separately flexed/extended. In addition the thumb can also be adducted/abducted in a way similar to the human thumb. The wrist is capable of flexion/extension. The actuators of the LUCS Haptic Hand III are controlled via an SSC-32 (Lynxmotion Inc.). The proprioceptive sensors are scanned with a MAX396CPI multiplexor chip and digitalized using an NiDaq 6008 (National Instruments). The NiDaq 6008 converts multiple analog input signals to digital signals, which are conveyed to the computer via a USB-port. The robot hand is equipped with two multiplexor chips, which means it is prepared for 21 additional sensors.

flexed/extended together. All this is similar to the human hand. The wrist can also be flexed/extended as the wrist of a human hand. The phalanges are made of plastic pipe segments and the force transmission from the actuators, which are located in the forearm, are handled by tendons inside the phalanges in a similar way to the tendons of a human hand. All fingers, except the thumb, are mounted

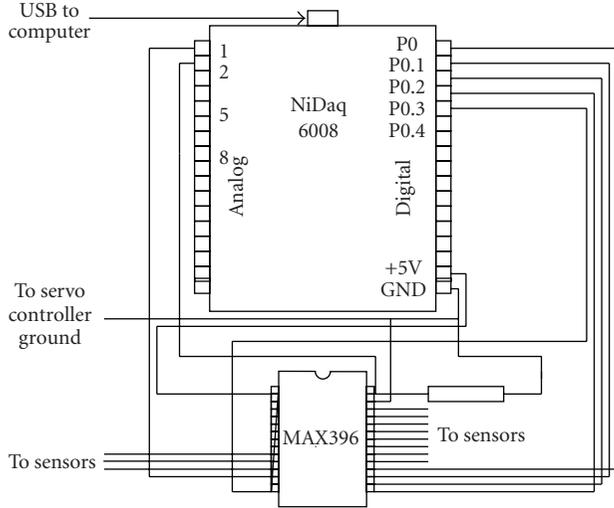


FIGURE 2: The circuits involved in the proprioceptive part of the LUCS Haptic Hand III. The NiDaq 6008 converts multiple analog input signals to digital signals that are conveyed to the computer via an, USB-port. The MAX396 chip is a multiplexor circuit for selection of sensor channels.

directly on the palm. The thumb is mounted on an RC servo, which enables the adduction/abduction. The RC servo is mounted on the proximal part of the palm, similar to the site of the thumb muscles in a human hand. The actuators of the fingers and the wrist are located in the forearm. This is also similar to the muscles that actuate the fingers of a human hand. The hand is actuated by in total 12 RC servos, and to get proprioceptive sensors, the internal potentiometers in the RC servos, except the RC servo that actuates the wrist, have been included in the sensory circuit (Figure 2). The resistances of these potentiometers are proportional to the angle of the different joints.

The software for the LUCS haptic hand III is developed in C++ and Java, and much of it runs within the Ikaros system [35, 36]. Ikaros provides an infrastructure for computer simulations of the brain and for robot control.

3. Self-Organizing ANNs

3.1. Self-Organizing Map. The SOM consists of an $I \times J$ grid of neurons with a fixed number of neurons and a fixed topology. Each neuron n_{ij} is associated with a weight vector $w_{ij} \in R^n$. During adaptation, the weight vectors for the neurons are adjusted to a degree which is determined by a neighbourhood function $N_{ijc}(t)$ with a size that decreases with time. The adaptation strength $\alpha(t)$ also decreases with time. The SOM variant used in our experiments is a dot product SOM with Gaussian neighbourhood. The adaptation algorithm works as follows.

At time t , each neuron n_{ij} receives an input vector $x(t) \in R^n$.

The neuron c associated with the weight vector $w_c(t)$ most similar to the input vector $x(t)$ is selected,

$$c = \arg \max_c \{ \|x(t)w_c(t)\| \}. \quad (1)$$

The weight vectors w_{ij} of the neurons n_{ij} are adapted according to:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)N_{ijc}(t)[x(t) - w_{ij}(t)], \quad (2)$$

where $0 \leq \alpha(t) \leq 1$ is the adaptation strength with $\alpha(t) \rightarrow 0$ when $t \rightarrow \infty$ and the neighbourhood function $N_{ijc}(t)$ is a Gaussian function the width of which decreases with time.

3.2. Growing Cell Structures. The GCS has a variable number of neurons and a k -dimensional topology, where k can be arbitrarily chosen. The adaptation of a weight vector in the GCS is done in a similar way as in the SOM, but the adaptation strength is constant over time and only the best matching unit and its direct topological neighbours are adapted. The GCS estimates the probability density function $p(x)$ of the input space by the aid of local signal counters that keep track of the relative frequencies of input signals gathered by each neuron. These estimates are used to indicate proper locations to insert new neurons. The insertion of new neurons by this method will result in a smoothing out of the relative frequencies between different neurons. The advantages of this approach are that the topology of the network will self-organize to fit the input space, the proper number of neurons for the network will be automatically determined and the learning rate and neighbourhood size parameters are constant over time. The basic building block and also the initial configuration of the GCS are a k -dimensional simplex. Such a simplex is for $k = 2$ a triangle. The variant of the GCS algorithm used in our experiments works as follows.

The network is initialized to contain $k + 1$ neurons with weight vectors $w_i \in R^n$ randomly chosen. The neurons are connected so that a k -dimensional simplex is formed.

At time step t , an input vector $x(t) \in R^n$ activates a winner neuron c for which the following is valid

$$c = \arg \min_c \{ \|x(t) - w_c(t)\| \}, \quad (3)$$

where $\|\cdot\|$ is the Euclidean distance, and the squared distance between the input vector and the weight vector of the winner neuron c is added to a local error variable E_c :

$$\Delta E_c = \|x(t) - w_c(t)\|^2. \quad (4)$$

The weight vectors are updated by fractions ε_b and ε_n , respectively, according to:

$$\Delta w_c(t) = \varepsilon_b(x(t) - w_c(t)), \quad (5)$$

$$\forall i \in N_c : \Delta w_i(t) = \varepsilon_n(x(t) - w_i(t)),$$

where N_c is the set of direct topological neighbours of c .

A neuron is inserted if the number of input vectors that have been generated so far is an integer multiple of a parameter λ . This is done by finding the neuron q with the largest accumulated error and the neuron f among its direct topological neighbours which has the weight vector with the longest distance from the weight vector of the neuron q , insert the new neuron r in between, remove the earlier connection (q, f) , and connect r with q and f and with all direct topological neighbours that are common for q and f . The weight vector for r is interpolated from the weight vectors for q and f :

$$w_r = \frac{w_q + w_f}{2}. \quad (6)$$

The local error counters for all neighbours to r are decreased by a fraction α that depends on the number of neighbours of r :

$$\forall i \in N_r : \Delta E_i = \left(-\frac{\alpha}{|N_r|} \right) \cdot E_i. \quad (7)$$

The error variable for r is set to the average of its neighbours:

$$E_r = \left(\frac{1}{|N_r|} \right) \cdot \sum_{i \in N_r} E_i, \quad (8)$$

and then the error variables of all neurons are decreased:

$$\forall i : \Delta E_i = -\beta E_i. \quad (9)$$

In GCS-DN, a neuron (or several if that is necessary to keep a consistent topological structure of k -dimensional simplices) is deleted, provided that the network has reached its maximum size; at the same occasions new neurons are inserted. Thereafter, new neurons are inserted again according to the algorithm described above until the network has reached its maximum size again. This process is repeated a preset number of times, in our experiments 250 times.

3.3. Growing Grid. The GG can be seen as an incremental variant of the SOM. It consists of an $I \times J$ grid of neurons with a fixed topology but with I and J increasing with time as new rows and columns are inserted. In addition to a weight vector $w_{ij} \in R^n$, each neuron n_{ij} also has a local counter variable T to estimate where to insert new rows or columns of neurons in the grid. The self-organizing process of a GG is divided into two phases: a growth phase and a fine-tuning phase. During the growth phase, the grid grows by insertion of new rows and columns until the wanted size of the network has been achieved. During the fine-tuning phase, the network size does not change and a decreasing adaptation strength $\alpha(t)$ is used. The size of the neighbourhood is not decreasing with time. Instead the network is growing with a constant neighbourhood size and therefore the fraction of all neurons that are adapted decreases over time. The variant of the GG algorithm used in our experiments is described below.

Growth Phase. Initialize the network to contain 2×2 neurons with weight vectors randomly chosen.

At time t , an input vector $x(t) \in R^n$ is generated and received by each neuron n_{ij} in the grid.

The neuron c associated with the weight vector $w_c(t)$ most similar to the input vector $x(t)$ is selected:

$$c = \arg \max_c \{ \|x(t)w_c(t)\| \}. \quad (10)$$

Increment the local counter variable T_c for c :

$$T_c = T_c + 1. \quad (11)$$

The weight vectors w_{ij} of the neurons n_{ij} are adapted according to:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha N_{ijc} [x(t) - w_{ij}(t)], \quad (12)$$

where $0 \leq \alpha \leq 1$ is the adaptation strength and the neighbourhood function N_{ijc} is a Gaussian function. Notice that α and N_{ijc} are not functions of t though.

A new row or column is inserted if the number of input vectors that have been generated so far is an integer multiple λ of the current number of neurons in the grid. This is done by finding the neuron q with the largest value of the local counter variable T and the neuron f among its direct topological neighbours which has the weight vector with the longest distance from the weight vector of the neuron q . Depending on the relative positions of q and f , a new row or a new column is inserted.

If q and f are in the same row, then a new column is inserted between the columns of q and f . The weight vectors for the new neurons are interpolated from their direct neighbours in the same row.

If q and f are in the same column, then a new row is inserted between the rows of q and f . The weight vectors for the new neurons are interpolated from their direct neighbours in the same column.

Adjust I or J to reflect the new numbers of rows and columns in the grid. Reset all local counter values:

$$T_{n_{ij}} = 0. \quad (13)$$

If the desired network size has not been reached, then go to step 2, that is, generate a new input vector.

Fine-Tuning Phase. This phase is similar to the growth phase but the adaptation strength $\alpha(t)$ is now decreasing with time and no insertions of new rows or columns are done. This phase stops after a preset number of iterations.

4. Proprioception-Based Systems

All the four systems (Figure 3) consist of the LUCS Haptic Hand III, sensory and motor drivers, a commander module that executes the grasping movements, and a Self-Organizing ANN (SO-ANN). The kind of SO-ANN employed is the only thing that distinguishes one system from another. The sensory driver scans the proprioceptive sensors when requested to do so by the commander module, while the

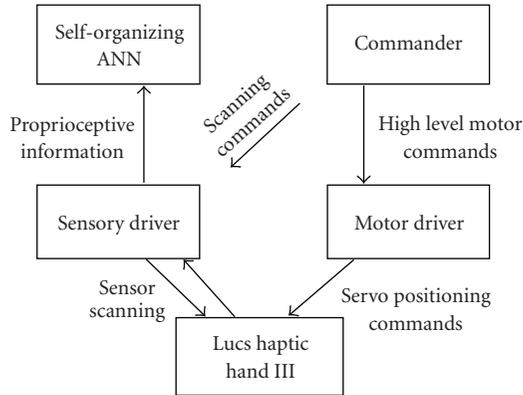


FIGURE 3: Schematic depiction of the systems. The commander module executes the grasps by sending high-level motor commands to the motor driver, which translates and conveys the information to the servo controller board of the robot hand. When the robot hand has become fully closed around the object, the commander module requests a scanning of the 11 proprioceptive sensors of the robot hand. The sensory information is conveyed as a vector to a Self-Organizing ANN (SO-ANN). The SO-ANN is a Self-Organizing-Map, a Growing Cell Structures, a Growing Cell Structures with Deletion of Neurons, or a Growing Grid.

motor driver translates high-level motor commands from the commander module to positioning commands for the robot hands servo controller board. When the commander executes a grasp, and the robot hand is fully closed around the object, the sensory driver scans the 11 proprioceptive sensors and outputs an eleven-element vector to the SO-ANN, which is adapted.

The SOM-based system uses a 225 neurons dot product SOM with plane topology, which uses softmax activation with the softmax exponent equal to 10 [37]. It is trained by 2000 iterations.

The GCS-based system grows, by inserting a new neuron every 19th iteration, until a size of 225 neurons has been reached.

The GCS-DN based system grows until a size of 225 neurons has been reached, also by inserting a new neuron every 19th iteration, then the deletion/insertion process described in Section 3.2 is repeated 250 times. Finally this yields a number of disconnected networks with altogether 225 neurons.

The GG-based system grows by inserting a new row or column each time the number of time steps t since the previous insertion equals a multiple λ of the current grid size, that is, until $t = \lambda IJ$ with $\lambda = 19$. The growth phase lasts until a minimum grid size of 225 neurons has been reached, then the model runs in fine tuning mode for 1000 iterations.

We have trained the systems with 10 objects (see Table 1 objects a–j). These objects are either cylinder shaped or block shaped. There are five objects of each shape category. All objects are sufficiently high to be of a nonvariable shape in those parts grasped by the robot hand, for example, a bottle is grasped on the part of equal diameter below the bottle neck.

During the grasping tests, the test objects were placed on a table with the open robot hand around them. If the objects were block shaped, we always placed the widest side against the palmar side of the robot hand.

To simplify the testing procedure, each object was grasped 5 times by the robot hand, that is, in total 50 grasps were carried out, and the sensory information was written to a file. Then the SO-ANN were trained and tested with this set of 50 samples. The training phase for the SOM system lasted for 2000 iterations. The GCS system was trained until a network size of 225 neurons was reached. The GCS-DN system was trained until a network size of 225 neurons was reached and then the insertion/deletion process described in Section 3.2 was repeated 250 times. The GG system was trained with a growth phase which lasted until the minimal network size reached 225 neurons, and then for 1000 iterations in fine tuning mode.

Each fully trained system was tested with the original training set and in addition with three new block-shaped and three new cylinder-shaped objects of variable sizes (see Table 1, objects 1–6) as described in the next section.

5. Generalization Tests

We have also tested if the systems were able to generalize their knowledge to new objects, that is, to objects not included in the training set. To this end we used 6 new objects, Table 1. 1–6, 3 cylinder shaped objects and 3 block shaped objects. The new objects were of variable sizes. The fully trained systems were fed by input from grasps of the new objects under the same conditions as the objects in the training set. Each object in the new set was grasped once and the activity in the SO-ANN for each system was recorded.

6. Results

The results are depicted in Figure 4. Figure 4(a) shows the centres of activation in the SOM in the fully trained SOM-based system when tested with the training set and the test set. The SOM seems to be organized according to shape. Four groups of objects can be distinguished in the map, large block shapes, small block shapes, large cylindrical shapes, and small cylindrical shapes. The SOM also seems to be organized in a clockwise manner according to size. The result of the generalization experiment shows that all test objects are mapped so that they are ordered according to size in the same way as the objects in the training set, and that they are also correctly mapped according to shape. The activations in the SOM also indicate that it is possible to discriminate individual object of the training set to a large extent and this is also true for the test objects, since each of the test objects is also mapped so that it can be identified as the most similar object of the training set. The results with the SOM-based system are thoroughly described in [29].

Figure 4(b) shows the centres of activation in the GCS in the fully trained GCS-based system. Only the part of the GCS which is activated by some object is shown in the figure. This system produces similar results as the SOM-based system,

TABLE 1: The 16 objects used in the experiments with the four systems. The objects a–j were used both for training and testing, whereas the objects 1–6 were used in the generalization tests.

| Label | Object | Shape | Size (mm) | Size (mm) |
|-------|----------------------|----------|---------------|------------|
| a | Tube | Cylinder | Diameter = 58 | — |
| b | Beer can | Cylinder | Diameter = 64 | — |
| c | Wood block | Block | Length = 75 | Width = 47 |
| d | Wine bottle | Cylinder | Diameter = 70 | — |
| e | Plastic block 1 | Block | Length = 63 | Width = 63 |
| f | Plastic bottle 2 | Cylinder | Diameter = 72 | — |
| g | Olive oil bottle | Block | Length = 65 | Width = 65 |
| h | Plastic bottle 1 | Cylinder | Diameter = 80 | — |
| i | Plastic block 2 | Block | Length = 80 | Width = 63 |
| j | Coffee package | Block | Length = 97 | Width = 67 |
| 1 | Card board package 1 | Block | Length = 77 | Width = 66 |
| 2 | Card board package 2 | Block | Length = 84 | Width = 62 |
| 3 | Card board package 3 | Block | Length = 95 | Width = 62 |
| 4 | Spice bottle | Cylinder | Diameter = 57 | — |
| 5 | Treacle bottle | Cylinder | Diameter = 63 | — |
| 6 | Plastic bottle 3 | Cylinder | Diameter = 79 | — |

that is, the organization of the GCS separates large block shapes, small block shapes, large cylinder shapes, and small cylinder shapes. The GCS is also organized according to size with the smallest objects represented uppermost in the GCS and the largest in the lowermost part. The ability for discrimination of individual objects is approximately similar as that for the SOM-based system. Also this system activates neurons at proper locations when fed with the objects of the generalization test set.

Figure 4(c) shows the final network structure of the fully trained GCS-DN based system. As can be seen, this network structure consists of several disconnected subnetworks. This is due to the removal of neurons that represent parts of the input space with a low value of the probability density function. As a result, such a network tends to self-organize into subnetworks that represent different clusters in the input space. This is also what happened in our experiments. As indicated in the figure, one or more subnetworks can be seen as representing one of the categories large block shapes, small block shapes, large cylinder shapes, and small cylinder shapes. The objects of the generalization test set activate neurons in the proper subnetworks except in one case, namely, the test object 1 is a large block but is identified as a large cylinder.

Figure 4(d) shows the centres of activation in the GG in the fully trained GG-based system. This system produces similar results as the SOM-based system and the GCS-based system, that is, the organization of the GG separates large block shapes, small block shapes, large cylinder shapes, and small cylinder shapes. As indicated in the figure, the GG is also organized according to size. The ability for discrimination of individual objects is approximately similar as that for the SOM-based system. All 6 objects of the generalization test set are mapped so that they can be

associated with the correct shape category and identified with the most similar object of the training set.

7. Discussion

We have experimented with four self-organizing systems for clustering of proprioceptive data collected by our anthropomorphic robot hand, the LUCS Haptic Hand III. All four systems were able to cluster the sensory information according to shape, and all four of them resulted in networks which preserve the size ordering of the training objects. The systems could also discriminate individual objects, more or less. The systems have proven to have an excellent generalization capacity. This is clearly illustrated in the categorization of the 6 new objects that offered different characteristics of shape and size.

The results are interesting because they reveal that the proprioceptive information encompasses information about both the shape and the size of the grasped objects, and in addition information that enables discrimination of the individual objects to some extent.

In comparison with our earlier systems for haptic shape perception [22–25], the current systems have turned out to be much more able to correctly categorize objects according to shape in a much wider size range, and this is done with a less computationally expensive model. The current systems were also able to map the sizes of the objects in an ordered fashion, and to discriminate between objects as long as they were not too similar. A human would probably have a similar problem if she, like our systems, was not able to detect the material properties of the objects or expressed differently, if all object were of exactly the same material and weight.

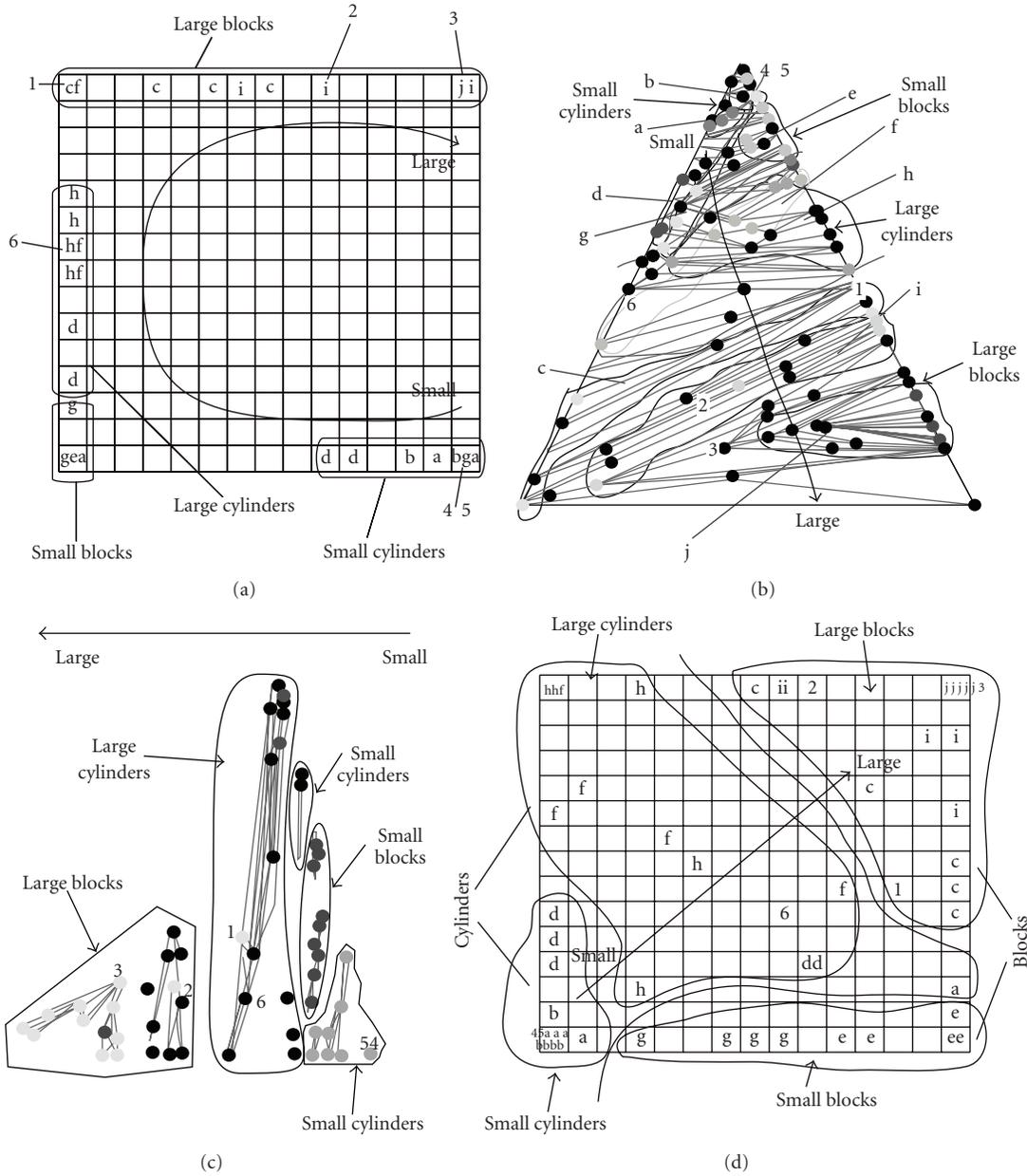


FIGURE 4: The test results of the four systems. (a) The SOM-based system is organized according to shape and size. Groups of large blocks, small blocks, large cylinders, and small cylinders can be distinguished. The activations tend to be located according to size of the objects in a clockwise manner. Individual objects can be discriminated to a large extent. (b) The GCS-based system produces similar results as the SOM-based system, that is, it is organized according to shape and size. The GCS-based system separates large blocks, small blocks, large cylinders, and small cylinders, and the objects are represented according to size with the smallest objects uppermost and the largest lowermost in the GCS. Also individual objects can be discriminated to a large extent. (c) The GCS-DN based system self-organized into subnetworks, where one or more subnetworks represent the categories large blocks, small blocks, large cylinders, and small cylinders. (d) The GG-based system separates large blocks, small blocks, large cylinders, and small cylinders, and the grid is organized according to size. Individual objects can be discriminated to a large extent. The 6 test objects (indicated with the numbers 1–6) not included in the training set activated neurons at proper locations perfectly in all systems but the GCS-DN based system. In that system, object 1 triggered activation in the wrong subnetwork. (see Table 1 for the meaning of the labels).

The SOM-based, the GCS-based, and the GG-based systems performed at approximately a similar level. This could be an argument for using the alternative neural network architectures GCS and GG instead of the SOM, because that reduces the number of parameters that have

to be set. According to Fritzke [38], the performance of the GCS is actually slightly better than the performance of the SOM in complex and realistic problems. The results of our experiments in [39] also point in that direction.

The GCS and the GCS-DN also have the virtue to get organized into networks whose topology reflect the probability density function of the input space. The GCS-DN is especially interesting since it has the property to automatically form disconnected subnetworks that represent clusters in the input space. It should be possible to implement an online version of the GCS-DN algorithm that never stops and that should result in a set of networks, that reflects the probability density function of the input space, which changes if the probability density function happens to be nonstationary. In other words, if the probability density function of the input space changed, then the set of subnetworks would change by the deletion of some subnetworks and the split, followed by growth of others.

It should be mentioned that the graphical presentation of GCS and GCS-DN could be improved. Fritzke [32] suggests a method on how to embed these kinds of networks in the plane for better visualizations. In this method, a physical model is maintained where the neurons are considered as discs influenced by attractive and repulsive forces.

The success with the GCS-based, the GCS-DN based, and the GG-based systems suggests an increased focus on our part on these kinds of self-organizing neural networks. The advantage of getting rid of several parameter settings like network size, learning rate, and neighbourhood settings can be important to succeed with more complex cognitive models with several coupled neural networks at multiple levels. To be forced to set all the parameters in a good way for all included neural networks with complex dependencies in such a model could prove to be overwhelming.

It would be interesting to compare our systems to self-organizing systems developed by others. Heidemann and Schöpfer [11] describe a haptic system, which consists of a plate with a touch sensitive array mounted on a robot arm. The system explores an object by sequences of contacts and feeds a self-organizing neural architecture with input. The system was able to learn to recognize 7 different objects when tested.

Natale and Torres-Jara [14] describe a system consisting of an upper body humanoid robot with a hand equipped with dome-like tactile sensors, which are sensitive to pressure from all directions, as well as position sensors (proprioception). The system also includes a camera together with a visual system for coarse localization of the object. The information gathered by the system was used as input to a SOM. When evaluated with 4 different objects, a bottle, a box, and two cups, these objects were mapped differently. However, the cups could not be distinguished from each other.

When compared with the two systems described above our current systems stand out in that they are able to categorize the objects according to shape, order them according to size, as well as recognize individual objects to a large extent.

In the future, we plan to increase the use of neural networks like GCS and GG as an alternative to the SOM in our haptic systems. By doing so, we will reduce the number of parameters that have to be set explicitly and this should yield more robust systems.

Because of the successful approach with using proprioceptive information as a base for haptic shape perception as well as size perception, we will in the nearest future continue our research in haptic perception with the following task: try to bring the proprioceptive systems to their absolute limits, for example, by exploiting the possibility of the LUCS Haptic Hand III to carry out a more active exploration than simply grasping the objects in only one way. This can be done by adducting/abducting the thumb and by flexing/extending the wrist differently in different grasps

At a later stage, we will study the interaction between haptics and vision. This would be interesting because these modalities interact to a considerable extent [40].

Acknowledgment

The authors want to acknowledge the financial support from Stiftelsen Landshövding Per Westlings Minnesfond to the LUCS Haptic Hand III.

References

- [1] P. Dario, C. Laschi, A. Menciassi, E. Guglielmelli, M. C. Carrozza, and S. Micera, "Interfacing neural and artificial systems: from neuroengineering to neurorobotics," in *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, pp. 418–421, 2003.
- [2] K. J. DeLaurentis and C. Mavroidis, "Development of a shape memory alloy actuated robotic hand," 2000, <http://citeseer.ist.psu.edu/383951.html>.
- [3] C. Rhee, W. Chung, M. Kim, Y. Shim, and H. Lee, "Door opening control using the multi-fingered robotic hand for the indoor service robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 4011–4016, New Orleans, La, USA, 2004.
- [4] H. Sugiuchi, Y. Hasegawa, S. Watanabe, and M. Nomoto, "A control system for multi-fingered robotic hand with distributed touch sensor," in *Proceedings of the 26th Annual Conference of the IEEE Industrial Electronics Society (IECON '00)*, vol. 1, pp. 434–439, Nagoya, Japan, October 2000.
- [5] M. A. Arbib, A. Billard, M. Iacoboni, and E. Oztop, "Synthetic brain imaging: grasping, mirror neurons and imitation," *Neural Networks*, vol. 13, no. 8-9, pp. 975–997, 2000.
- [6] A. H. Fagg and M. A. Arbib, "Modeling parietal-premotor interactions in primate control of grasping," *Neural Networks*, vol. 11, no. 7-8, pp. 1277–1303, 1998.
- [7] P. K. Allen and P. Michelman, "Acquisition and interpretation of 3-D sensor data from touch," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 397–404, 1990.
- [8] S. Caselli, C. Magnanini, and F. Zanichelli, "Haptic object recognition with a dextrous hand based on volumetric shape representations," in *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI '94)*, pp. 280–287, Las Vegas, Nev, USA, 1994.
- [9] P. Dario, C. Laschi, M. C. Carrozza, et al., "An integrated approach for the design and development of a grasping and manipulation system in humanoid robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 1–7, Takamatsu, Japan, October 2000.
- [10] I. Erkmen, A. M. Erkmen, A. E. Tekkaya, and T. Pasinioglu, "Haptic perception of shape and hollowness of deformable

- objects using the anthrobot-III robot hand,” *Journal of Robotic Systems*, vol. 16, no. 1, pp. 9–24, 1999.
- [11] G. Heidemann and M. Schöpfer, “Dynamic tactile sensing for object identification,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA ’04)*, vol. 1, pp. 813–818, New Orleans, La, USA, 2004.
- [12] K. Hosoda, Y. Tada, and M. Asada, “Anthropomorphic robotic soft fingertip with randomly distributed receptors,” *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 104–109, 2006.
- [13] J. Jockusch, J. Walter, and H. Ritter, “A tactile sensor system for a three-fingered robot manipulator,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3080–3086, Albuquerque, NM, USA, 1997.
- [14] L. Natale and E. Torres-Jara, “A sensitive approach to grasping,” in *Proceedings of the 6th International Workshop on Epigenetic Robotics*, pp. 87–94, 2006.
- [15] E. M. Petriu, S. K. S. Yeung, S. R. Das, A.-M. Cretu, and H. J. W. Spoelder, “Robotic tactile recognition of pseudorandom encoded objects,” *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 5, pp. 1425–1432, 2004.
- [16] S. A. Stansfield, “A haptic system for a multifingered hand,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 658–664, Sacramento, Calif, USA, April 1991.
- [17] D. Taddeucci, C. Laschi, R. Lazzarini, R. Magni, P. Dario, and A. Starita, “An approach to integrated tactile perception,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA ’97)*, vol. 4, pp. 3100–3105, Albuquerque, NM, USA, April 1997.
- [18] M. Johnsson, “LUCS haptic hand I,” Tech. Rep. 8, LUCS Minor, Lund University Cognitive Studies, 2004.
- [19] M. Johnsson, R. Pallbo, and C. Balkenius, “Experiments with haptic perception in a robotic hand,” in *Advances in Artificial Intelligence in Sweden*, pp. 81–86, Mälardalen University, 2005.
- [20] M. Johnsson, R. Pallbo, and C. Balkenius, “A haptic system for the LUCS haptic hand I,” in *Proceedings of the 1st International Work-Conference on the Interplay between Natural and Artificial Computation (IWINAC ’05)*, vol. 3561 of *Lecture Notes in Computer Science*, pp. 386–395, Springer, 2005.
- [21] M. Johnsson and C. Balkenius, “Experiments with artificial haptic perception in a robotic hand,” *Journal of Intelligent and Fuzzy Systems*, vol. 17, no. 4, pp. 377–385, 2006.
- [22] M. Johnsson and C. Balkenius, “LUCS haptic hand II,” Tech. Rep. 9, LUCS Minor, Lund University Cognitive Studies, 2006.
- [23] M. Johnsson and C. Balkenius, “Haptic perception with a robotic hand,” in *Proceedings of the 9th Scandinavian Conference on Artificial Intelligence (SCAI ’06)*, Espoo, Finland, 2006.
- [24] M. Johnsson and C. Balkenius, “A robot hand with T-MPSOM neural networks in a model of the human haptic system,” in *Proceedings of Towards Autonomous Robotic Systems (TAROS ’06)*, pp. 80–87, Surrey University, Guildford, UK, 2006.
- [25] M. Johnsson and C. Balkenius, “Neural network models of haptic shape perception,” *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 720–727, 2007.
- [26] M. Johnsson and C. Balkenius, “Recognizing texture and hardness by touch,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’08)*, pp. 482–487, Nice, France, 2008.
- [27] M. Johnsson and C. Balkenius, “Associating SOM representations of haptic submodalities,” in *Proceedings of Towards Autonomous Robotic Systems (TAROS ’08)*, pp. 124–129, University of Edinburgh, Edinburgh, UK, 2008.
- [28] T. Kohonen, *Self-Organization and Associative Memory*, Springer, Berlin, Germany, 1988.
- [29] M. Johnsson and C. Balkenius, “Experiments with proprioception in a self-organizing system for haptic perception,” in *Proceedings of Towards Autonomous Robotic Systems (TAROS ’07)*, pp. 239–245, University of Wales, Aberystwyth, UK, 2007.
- [30] M. Johnsson and C. Balkenius, “LUCS haptic hand III—an anthropomorphic robot hand with proprioception,” Tech. Rep. 13, LUCS Minor, 2007.
- [31] M. Johnsson, D. Gil Mendez, and C. Balkenius, “Touch perception with SOM, growing cell structures and growing grids,” in *Proceedings of Towards Autonomous Robotic Systems (TAROS ’08)*, pp. 79–85, University of Edinburgh, Edinburgh, UK, 2008.
- [32] B. Fritzke, “Growing cell structures—a self-organizing network for unsupervised and supervised learning,” *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1994.
- [33] B. Fritzke, “Unsupervised ontogenic networks,” in *Handbook of Neural Computation*, E. Fiesler and R. Beale, Eds., IOP Publishing and Oxford University Press, London, UK, 1997.
- [34] B. Fritzke, “Growing Grid—a self-organizing network with constant neighborhood range and adaptation strength,” *Neural Processing Letters*, vol. 2, no. 5, pp. 9–13, 1995.
- [35] C. Balkenius and J. Morén, “From isolated components to cognitive systems,” *ERCIM News*, p. 16, April 2003.
- [36] C. Balkenius, J. Morén, and B. Johansson, *Building System-Level Cognitive Models with Ikaros*, Lund University Cognitive Studies, Lund, Sweden, 2007.
- [37] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, NY, USA, 1995.
- [38] B. Fritzke, “Kohonen feature maps and growing cell structures—a performance comparison,” in *Proceedings of the Advances in Neural Information Processing Systems (NIPS ’92)*, pp. 123–130, Denver, Colo, USA, 1992.
- [39] D. Gil Mendez, M. Johnsson, A. Soriano Paya, and D. Ruiz Fernandez, “Artificial neural networks for diagnoses of dysfunctions in urology,” in *Proceedings of the 1st International Conference on Health Informatics (HEALTHINF ’08)*, vol. 2, pp. 191–196, Madeira, Portugal, 2008.
- [40] U. Castiello, “The neuroscience of grasping,” *Nature Reviews Neuroscience*, vol. 6, no. 9, pp. 726–736, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

