Touch Perception with SOM, Growing Cell Structures and Growing Grids

Magnus Johnsson, David Gil Mendez and Christian Balkenius

***Abstract*— We have implemented four bio-inspired self-organizing haptic systems based on proprioception on a 12 d.o.f. anthropomorphic robot hand. 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 both the training set as well as in the generalization experiments, i.e. they mapped the objects according to shape and size and discriminated individual objects. The GCS-DN system managed to evolve disconnected networks representing 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.**

I. INTRODUCTION

When designing a neural network based 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) [18] that we have used in previous work. This is often a very good choice but it suffers from some limitations, e.g. 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 explored and compared three alternative neural network architectures that avoid some or all of these problems and compared them with the SOM in the context of a proprioception based haptic system. These alternative neural network architectures are the Growing Cell Structures (GCS) and the Growing Cell Structures with Deletion of Neurons (GCS-DN) [3][5], and the Growing Grid (GG) [4].

The choice of a haptic perception system as the context for these explorations is due to our extensive experience in this area with investigations of haptic size perception systems [7][8][9][10], of haptic shape perception systems [11][12][13][14], of proprioception based haptic shape/size perception systems [15], and of haptic texture/hardness perception systems [17]. We also have experience in the use of GCS [6] and we have access to a suitable robot platform consisting of an anthropomorphic robot hand, the LUCS Haptic Hand III [16] which has been employed in the experiments for this paper. However, our findings should be equally applicable in systems mimicking other sensory modalities.

The LUCS Haptic Hand III is a five fingered 12 dof anthropomorphic robot hand with 11 proprioceptive sensors (Fig. 1). The wrist can be flexed/extended. The thumb has two and the other fingers have three phalanges. The thumb can be adducted/abducted and separately flexed/extended in the proximal and the distal joints. The other fingers can be separately flexed/extended in their proximal joints whereas the middle and the distal joints are flexed/extended together. This is similar to the human hand. The phalanges are made of plastic pipe segments and the force transmission from the actuators in the forearm are handled by tendons inside the phalanges. All fingers, except the thumb, are mounted directly on the palm. The thumb is mounted on a RC servo, which enables the adduction/abduction. This RC servo is mounted proximally on the palm. Internal potentiometers in the actuators are used as proprioceptive sensors. The resistances of these potentiometers are proportional to the angles of the different joints.

This paper compares four similar bio-inspired haptic size/shape perception systems based on proprioception and the anthropomorphic robot hand LUCS Haptic Hand III. The 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 GCS, the third uses the GCS-DN and the fourth uses the GG.

II. SELF-ORGANIZING ANNS

A. Self-Organizing Map

The SOM consists of a $I \times J$ 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:

Magnus Johnsson is with the Department of Computer Science and Lund University Cognitive Science, Lund University, Sweden Magnus.Johnsson@lucs.lu.se

David Gil Mendez is with Computing Technology and Data Processing, University of Alicante, Spain dgil@dtic.ua.es

Christian Balkenius is with Lund University Cognitive Science, Lund University, Sweden Christian.Balkenius@lucs.lu.se

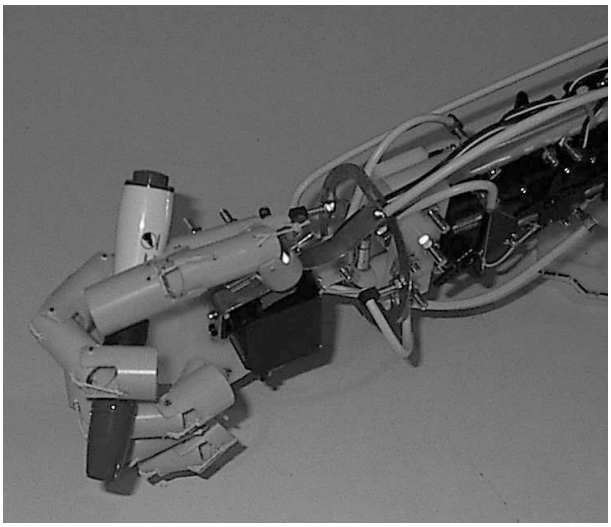

Fig. 1. *The LUCS Haptic Hand III while holding a pen. The five-fingered 12 dof robot hand is closely modelled after the human hand with respect to size and the proportions between its parts. The actuators of the robot hand are located in the forearm, which can also be seen in the figure, and tendons are used for force transmission. The robot hand is equipped with a flexible wrist. Each finger can be separately flexed/extended in the proximal joint, whereas the medial and distal joints are flexed/extended together similar to human fingers. The thumb can be adducted/abducted and it has a proximal and a distal phalange that can be separately flexed/extended. Variable potentiometers internal to the actuators are used as proprioceptive sensors, which are scanned with a MAX396CPI multiplexor chip and digitalized using a 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 actuators of the LUCS Haptic Hand III are controlled via a SSC-32 (Lynxmotion Inc.).*

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:

$$c = \arg\,\max_c\{||x(t)w_c(t)||\} \tag{1}$$

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

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

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

B. 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 is 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 is 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 $\mathrm{w}_i \in R^n$ 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:

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

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:

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

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

$$\Delta w_c(t) = \varepsilon_b(x(t) - w_c(t)) \tag{5}$$

$$\forall i \in N_c : \Delta w_i(t) = \varepsilon_n(x(t) - w_i(t)), \tag{6}$$

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:

$$w_r = (w_q + w_f)/2. \tag{7}$$

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

$$\forall i \in N_r : \Delta E_i = (-\alpha/|N_r|) \cdot E_i, \tag{8}$$

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

$$E_r = (1/|N_r|) \cdot \sum_{\iota \in Nr} E_i, \quad (9)$$

and then the error variables of all neurons are decreased:

$$\forall i : \Delta E_i = -\beta E_i \quad (10)$$

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.

C. 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:

$$c = \arg\max_c \{||x(t)w_c(t)||\} \quad (11)$$

Increment the local counter variable T_c for c:

$$T_c = T_c + 1 \quad (12)$$

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

$$w_{ij}(t+1) = w_{ij}(t) + \alpha N_{ijc}\,[x(t) - w_{ij}(t)] \quad (13)$$

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:

$$T_{n_{ij}} = 0 \quad (14)$$

If desired network size has not been reached, then go to step 2, i.e. 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.

III. MODELS

The models differ in one respect, namely the kind of self-organizing neural network employed. The models consist of the LUCS Haptic Hand III, sensory and motor drivers, a Self-Organizing Neural Network (SO-ANN) and a commander module that executes the grasping movements (Fig. 2). The sensory driver scans the proprioceptive sensors when requested to do so by the commander module, while the motor driver translates high level motor commands from the commander module to commands appropriate 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-elements vector to the SO-ANN, which is adapted.

The SOM model uses a 225 neurons dot product SOM with plane topology, which uses softmax activation with the softmax exponent equal to 10 [1]. It is trained by 2000 iterations.

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

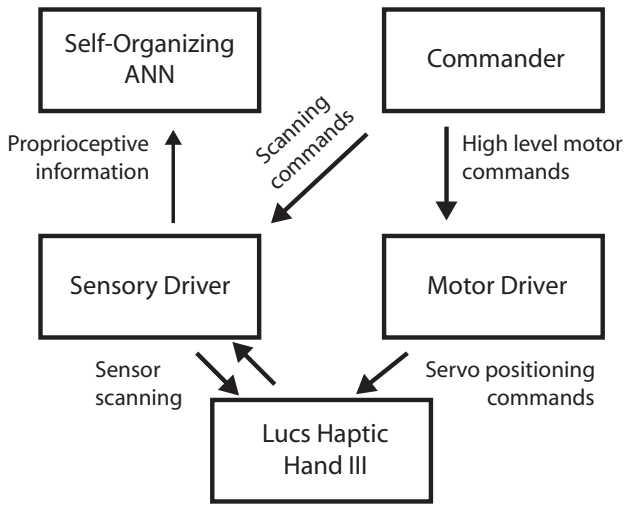

Fig. 2. *Schematic depiction of the models. The commander module executes the grasps by sending high-level motor commands to the motor driver, which translates and conveys the information to the servo controller board of the robot hand. When the robot hand has become fully closed the commander module request 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, or a Growing Cell Structures, or a Growing Cell Structures with Deletion of Neurons, or a Growing Grid.*

The GCS-DN model 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 2.2 is repeated 250 times. Finally this yields a number of disconnected networks with altogether 225 neurons.

The GG model 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, i.e. 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.

IV. TESTING THE MODELS

We have trained the models 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 non-variable shape in those parts grasped by the robot hand, e.g. 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 longest side against the palmar side of the robot hand.

To simplify the testing procedure each object was grasped 5 times by the robot hand, i.e. in total 50 grasps were carried out, and the sensory information were written to a file. Then the SO-ANN were trained and tested with this set of 50 samples. The training phase for the SOM model lasted for 2000 iterations. The GCS model was trained until a network size of 225 neurons was reached. The GCS-DN model was trained until a network size of 225 neurons was reached and then the insertion/deletion process described in section 2.2 was repeated 250 times. The GG model 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 model 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).

V. RESULTS

The results are depicted in Fig. 3. Fig 3A shows the centres of activation in the SOM in the fully trained SOM model 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 model are thoroughly described in [15].

Fig 3B shows the centres of activation in the GCS in the fully trained GCS model. Only the part of the GCS which is activated by some object is shown in the figure. This model produces similar results as the SOM model, i.e. 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 model. Also this model activates neurons at proper locations when fed with the objects of the generalization test set.

Fig 3C shows the final network structure of the fully trained GCS-DN model. 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,

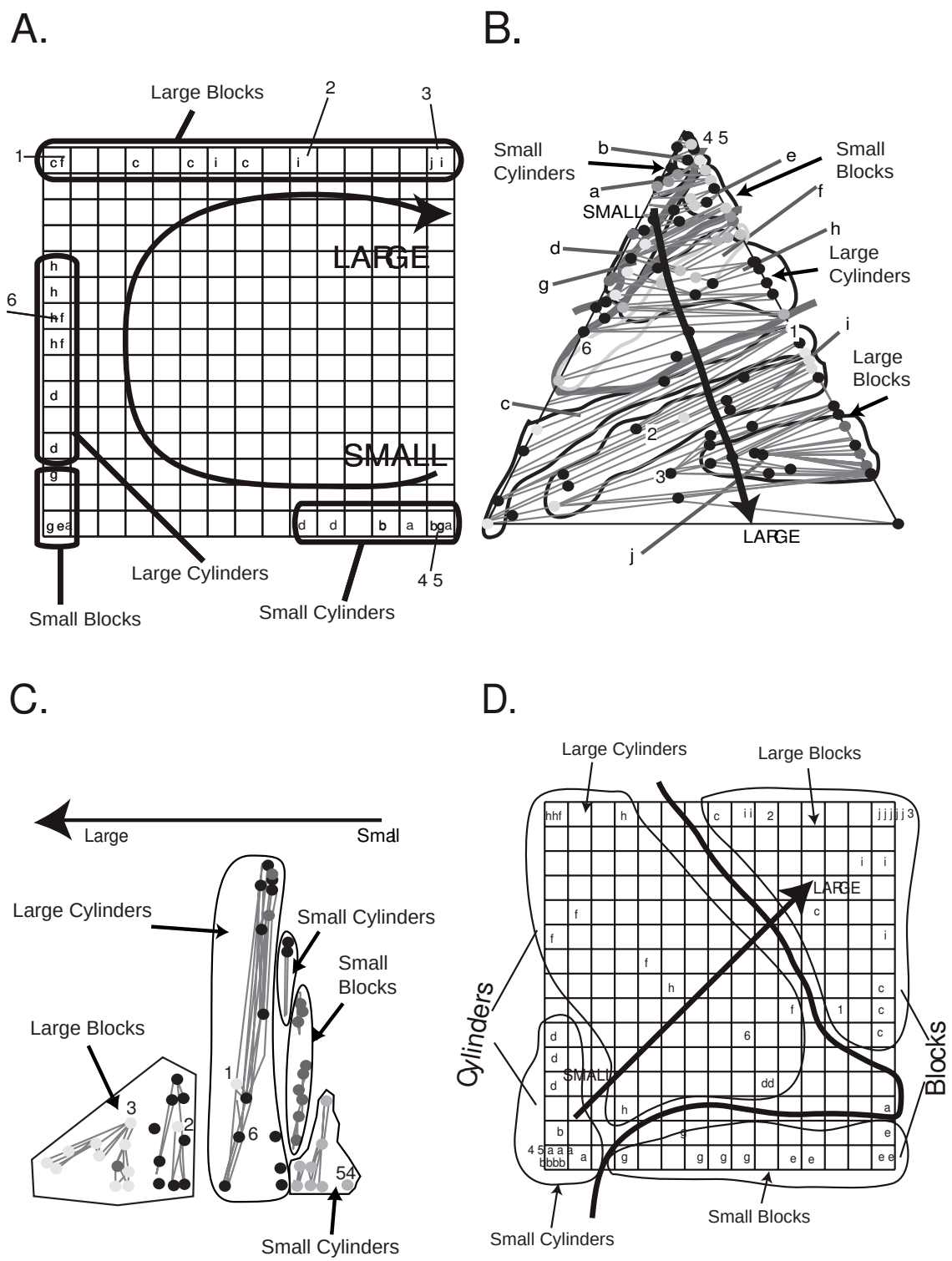

Fig. 3. *The test results of the four models. A: The SOM model is organized according to shape and size. Groups of large blocks, small blocks, large cylinders and small cylinders can be distinguished. The activations tend to be located according to size of the objects in a clockwise manner. Individual objects can be discriminated to a large extent. B: The GCS model produces similar results as the SOM model, i.e. it is organized according to shape and size. The GCS model separate 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 model self-organized into sub networks, where one or more sub networks represent the categories large blocks, small blocks, large cylinders and small cylinders. D: The GG model separate 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 models but the GCS-DN model. In that model object 1 triggered activation in the wrong subnetwork. (See Table 1 for the meaning of the labels).*

TABLE I

The 16 objects used in the experiments with the four models. 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	-

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.

Fig 3D shows the centres of activation in the GG in the fully trained GG model. This model produces similar results as the SOM model and the GCS model, i.e. 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 model. 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.

VI. DISCUSSION

We have experimented with four self-organizing models for clustering of proprioceptive data collected by our anthropomorphic robot hand, the LUCS Haptic Hand III. All four models were able to cluster the sensory information according to shape, and all four of them resulted in networks which preserves the size ordering of the training objects. The models 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 SOM, the GCS and the GG performed at approximately a similar level. This could be an argument for using the alternative neural network architechtures GCS and GG instead of the SOM, because that reduces the number of parameters that have to be set. According to Fritzke [2] 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 [6] also points 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 especielly 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 non-stationary. 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 [3] 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, the GCS-DN and the GG 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.

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.

VII. ACKNOWLEDGMENTS

We want to acknowledge the support from Stiftelsen Landshövding Per Westlings Minnesfond for finacial support

to the LUCS Haptic Hand III.

REFERENCES

[1] C. M. Bishop. (1995). Neural Networks for Pattern Recognition. *Oxford University Press*, Oxford, New York.

[2] Fritzke, B. (1992). Kohonen Feature Maps and Growing Cell Structures - a Performance Comparision. *NIPS 1992*, Denver

[3] Fritzke, B. (1993). Growing Cell Structures - A Self-organizing Network for Unsupervised and Supervised Learning. *Neural Networks*, 7, 9, 1441-1460.

[4] Fritzke, B. (1995). Growing Grid - a self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 2, 5,

[5] Fritzke, B. (1997). Unsupervised ontogenic networks. *In Fiesler and Beale (eds.). Handbook of Neural Computation*, IOP Publishing Ltd. and Oxford University Press.

[6] Gil Mendez, D., Johnsson, M., Soriano Paya, A., & Ruiz Fernandez, D. (2008). Artificial Neural Networks for Diagnoses of Dysfunctions in Urology. *Proceedings of Healthinf 2008*, Madeira, Portugal.

[7] Johnsson, M. (2004). LUCS Haptic Hand I - Technical Report, *LUCS Minor, Lund University Cognitive Studies - Technical Reports*, 8.

[8] Johnsson, M., Pallbo, R, & Balkenius, C. (2005a). Experiments with haptic perception in a robotic hand, *Advances in artificial intelligence in Sweden*, 81-86, Mälardalen University.

[9] Johnsson, M., Pallbo, R., & Balkenius, C. (2005b). A haptic system for the LUCS Haptic Hand I, *Proceedings of IWINAC 2005*, 338-397, Springer Verlag.

[10] Johnsson, M., & Balkenius, C. (2006a). Experiments with Artificial Haptic Perception in a Robotic Hand, *Journal of Intelligent and Fuzzy Systems*.

[11] Johnsson, M., & Balkenius, C. (2006b). LUCS Haptic Hand II, *LUCS Minor, Lund University Cognitive Studies - Technical Reports*, 9.

[12] Johnsson, M., & Balkenius, C. (2006c). Haptic Perception with a Robotic Hand, *Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006), Espoo, Finland.*

[13] Johnsson, M., & Balkenius, C. (2006d). A Robot Hand with T-MPSOM Neural Networks in a Model of the Human Haptic System, *Proceedings of TAROS 2006*, Surrey University, Guildford, UK, 80-87.

[14] Johnsson, M., & Balkenius, C. (2007a). Neural Network Models of Haptic Shape Perception, *Journal of Robotics and Autonomous Systems*, in press.

[15] Johnsson, M., & Balkenius, C. (2007b). Experiments with Proprioception in a Self-Organizing System for Haptic Perception, *Proceedings of TAROS 2007*, University of Wales, Aberystwyth, UK, 239-245.

[16] Johnsson, M., & Balkenius, C. (2007c). LUCS Haptic Hand III - An Anthropomorphic Robot Hand with Proprioception. *LUCS Minor 13.*

[17] Johnsson, M., & Balkenius, C. (2008). Recognizing Texture and Hardness by Touch. *To appear in the proceedings of IROS 2008*, Nice, France.

[18] Kohonen, T. (1988). *Self-Organization and Associative Memory*, Berlin Heidelberg, Springer-Verlag.