Haptic Perception with a Robotic Hand

Magnus Johnsson[⋆]
[⋆]Dept. of Computer Science and
Lund University Cognitive Science
Lund University, Sweden
Magnus.Johnsson@lucs.lu.se

Christian Balkenius[†]
[†]Lund University Cognitive Science
Lund University, Sweden
Christian.Balkenius@lucs.lu.se

Abstract

We have developed an 8 d.o.f. robot hand which has been tested with three computational models of haptic perception. Two of the models are based on the tensor product of different proprioceptive and sensory signals and a self-organizing map (SOM), and one uses a novel self-organizing neural network, the T-MPSOM. The computational models have been trained and tested with a set of objects consisting of hard spheres, blocks and cylinders. The first computational model, which is based on the tensor product, was able to discriminate between all the test objects. The second and third models could also discriminate between the test objects, but in addition they were capable of shape categorization.

1 Introduction

Haptic perception is an active tactile process that involves both sensory and motor systems to identify an object. In the human hand, somatosensory information is provided by receptors in the skin, the muscles and the joints. This information is needed to calculate the path of the hand, to control its locomotion and to enable an adequate grasp.

To some extent, objects can be identified with passive touch. In this case, the identification is based on the excitation of specific receptors sensitive to pressure, heat, touch and pain. The combination of these sensations provides information for object identification (Millar, 2006). A combination of cutaneous and proprioceptive information about the configuration of the hand is needed for better discrimination.

Surprisingly little work has addressed the problem of haptic perception. Most models of hand control have focused on the motor aspect rather than on haptic perception (Arbib, Billard, Iacoboni & Oztop, 2000; Fagg & Arbib, 1998). This is also true for the robotic hand research which has mainly looked at grasping and object manipulation (DeLaurentis & Mavroidis, 2000; Sugiuchi, Hasegawa, Watanabe & Nomoto, 2000; Dario, Laschi, Menciassi, Guglielmelli, Carrozza & Micera, 2003; Rhee, Chung, Kim, Shim & Lee, 2004). There are some exceptions however. Dario et al (2000) developed a system capable of haptic object classification. Another example is a system made for studies of the interaction between vision and haptics (Coelho, Piater & Grupen, 2001). Hosoda, Tada and Asada (2006) have built an anthropomorphic robot finger with a soft fingertip with randomly distributed embedded receptors. Jockusch, Walter and Ritter (1997) have developed a cost effective artificial fingertip with force/position sensors and slippage detection.

Heidemann and Schöpfer (2004) have developed a system for haptic object identification that uses a low-cost 2D pressure sensor with coarse resolution. The sensor is mounted on a PUMA-200 industrial robot arm. The system collects information by repeated contacts with the object. The collected information is combined is used as input to a three-step processing architecture that lets features form automatically.

Our previous research in haptics consists of the design and implementation of a simple three-fingered robot hand, the Lucs Haptic Hand I, together with a series of computational models (Johnsson, 2004, 2005; Johnsson et al., 2005a, 2005b; Johnsson & Balkenius, 2006a). This paper describes its successor, the Lucs Haptic Hand II, and three haptic models developed for it. The Lucs Haptic Hand II (Fig. 1) is an 8 d.o.f. three-fingered robot hand equipped with 45 pressure sensors developed at Lund University Cognitive Science (Johnsson & Balkenius, 2006b). Each finger consists of a proximal finger segment articulated against a triangular plastic plate, and a distal finger segment articulated against the proximal segment. Each finger segment contains a RC servos. A sensor plate is mounted on the inner side of each finger segment and contains 7 or 8 pressure sensitive sensors.

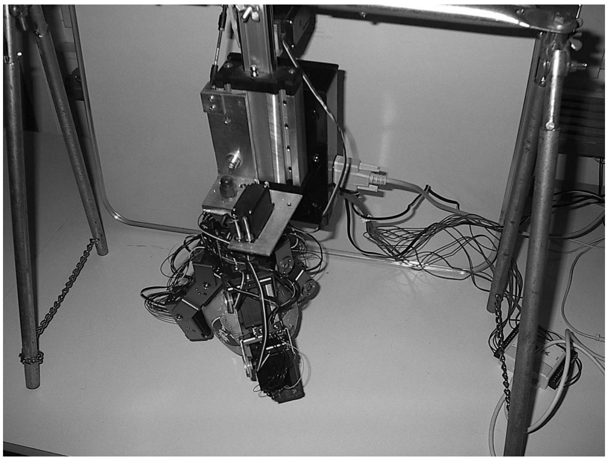

Figure 1: *The Lucs haptic hand II while grasping a soft ball. The 8-dof robot hand has three fingers, each consisting of two segments symmetrically mounted on a triangular plastic plate. The hand is equipped with a wrist and a lifting mechanism. The finger segments are built with RC servos mounted with servo brackets. All the actuators of the Lucs Haptic Hand II are controlled via a SSC-32 (Lynxmotion, Inc.). At each finger segment there is a plate mounted. The plates are equipped with push sensors.*

The triangular plastic plate is mounted on a wrist consisting of a bearing, and an actuator connected to a rod for force transmission. The wrist enables horizontal rotation of the robot hand. The wrist is in turn mounted on a lifting mechanism consisting of an actuator and a splint. Fig. 2 depicts the sensors status during a moment in a grasping movement. A movie showing the Lucs haptic hand II in a grasping task is available on the web site (Johnsson, 2005).

Model 1 uses the tensor product (outer product) to combine cutaneous and proprioceptive information gathered by the robot hand. The tensor product is an operation between a n-dimensional column vector $x = (x_1, \ldots, x_n)^T$ and a m-dimensional row vector $y = (y_1, \ldots, y_m)$ resulting in a $n \times m$ matrix M, where

$$\mathbf{M} = \begin{pmatrix} x_1y_1 & x_1y_2 & \cdots \\ x_2y_1 & x_2y_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Model 2 uses the tensor product to combine the tactile information in several steps. Model 3 is similar to Model 2 but substitutes the tensor product operations with Tensor Multiple Peak Self-Organizing Maps (T-MPSOM:s), a novel neural network architecture that combines the computations of the tensor product with the merits of self-organizing maps. The aim of these models was to take another step in our research and implement a system capable of haptic shape categorization.

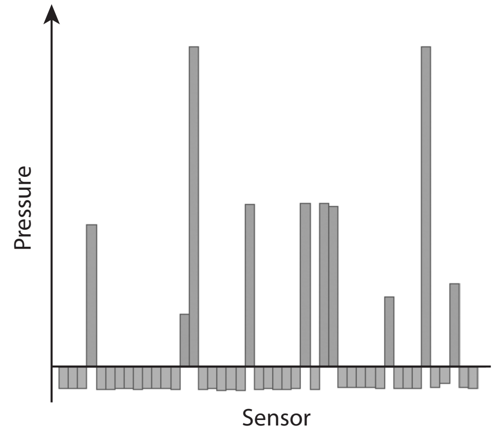

Figure 2: *Signals from the Lucs Haptic Hand II during the grasping of an object.*

2 Model 1

2.1 Design

The software for the Lucs haptic hand II is developed as Ikaros modules (Balkenius & Morén, 2003).

Beside the Lucs Haptic Hand II and the sensory and motor drivers, model 1 (Fig. 3) consists of four common Ikaros modules.

The Grasping Module takes care of the individual grasping movements, i.e. not the changes in the height of the robot hand or the angle of the wrist. When executing a grasping movement the module starts by moving the proximal finger segments. The sensor status is measured individually for each finger segment and a finger segment stops its movement when the total change of sensors registering exceeds a threshold or the position of the finger segment has reached a maximal allowed position. When the proximal segment of a finger stops, the distal segment starts to move. It continues to move until the total change of sensors registration for that particular segment exceeds a certain threshold or the position of the segment reaches a maximal position. The idea here is to let the robot hand take a shape that is in accordance with the grasped object. The Grasping Module controls the motor driver and also receives input from it. This input is a representation of the current configuration of the robot hand and can be thought of as proprioceptive information. The proprioceptive information is coded as a vector. This vector can be considered as a series of sequences of 10 elements, where each sequence code for the position of a moveable part of the Lucs Haptic Hand II. The position of a moveable part is coded by setting one of the elements in the sequence to 1 and the other elements to 0. In the second and the third models this vector is split into three, one that codes for the wrist angle, one that code for the height of the robot hand and one that code for the finger segments. The Grasping Module also receives information about the sensors status as input from the sensory driver. This tactile information is coded by a vector with 45 elements corresponding to the 45 sensors. Upwards in the model, the Grasping Module communicates with the Commander Module and with the STM Module. The input from the motor driver is combined with the input from the sensory driver by the use of tensor product between the proprioceptive vector and the sensory vector. This is calculated in the Grasping Module and is sent as output to the STM Module.

The Commander Module is responsible for the exploration of the object. This is done by carrying out a sequence of nine different grasps at two different heights and with 5 different wrist angles. In the model the haptic exploration is implemented by letting the Grasping Module receive orders about individual grasping movements and at what height and wrist angle the individual grasping movement is going to take place. The exploration continues until a sequence of nine exploration grasps have been executed with the actual object. The Commander Module receives signals from the Grasping Module about the status of the grasping movements, i.e. whether it is in progress or completed.

The STM (Short-Term Memory) Module receives the tensor product matrix from the Grasping Module, and the matrices from the whole exploration of an object are superimposed in the STM Module. Therefore the tactile information from the beginning to the end of the exploration of an object are put together and represented with one matrix. When haptically exploring an object the sensory information should, in raw or in refined form, be stored temporarily in the brain before the recognition of the object happens after some active exploration.

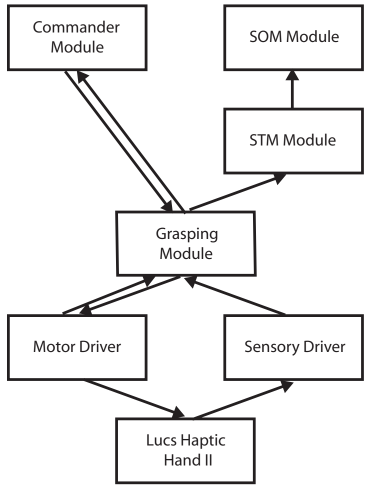

Figure 3: *Schematic depiction of model 1. The Lucs Haptic Hand II conveys the status of the sensors to the Sensory Driver which in turn conveys the status of the sensors as an array to the Grasping Module. The Motor Driver conveys, to the Lucs Haptic Hand II, the wanted positions of the servos in the robot hand and the wanted time to reach these positions. In addition, the Motor Driver conveys proprioceptive information, i.e. information about the configuration of the robot hand, the wrist and the lifting mechanism to the Grasping Module. The Grasping Module conveys the wanted configuration of the robot hand and the wanted time to reach it to the Motor Driver, the status of individual grasps to the Commander Module, and the Tensor product between proprioceptive and sensory information to the STM Module. The STM Module conveys a matrix of the tensor product superimposed from a whole exploration to the SOM Module. The Commander Module conveys orders to initiate grasping movements to the Grasping Module.*

The SOM Module When the exploration of the object is completed the output matrix of the STM Module that represents all tactile information from the exploration is used as input to the SOM Module. The SOM Module is a self-organizing neural network (Kohonen, 2001) with 225 neurons. In this module the learning and the categorization of this model takes place.

2.2 Grasping Tests

In order to test the model, we have used 6 different test objects, 2 different blocks, 2 different cylinders and 2 different spheres, see Table 1. To avoid a big influence of the hardness of the objects on the results, all the objects were made of hard materials like wood, hard plastic, hard board and metal.

To simplify the test procedure the system explored each of the 6 different objects 5 times, i.e. in total 30 explorations, and the output matrices from the STM Module were written to files. This set of 30 files was then used as a training and test set for the SOM Module. The model was trained with 1000 randomly chosen samples from the training set, then the trained model were tested with all 30 samples in the training set.

2.3 Results and Discussion

The result from the grasping test with this model is depicted in Fig. 4. In the figure the center of activity for each exploration sample in the training set is depicted. As can be seen the objects are categorized in a reproducible way. Also observe in the figure that the blocks in the lower part of the figure can be separated from other objects, i.e. from the spheres and the cylinders. An interpretation of this is that objects (in the training set) that are radial symmetric in at least one plane (the spheres and the cylinders) are separated from objects (in the training set) that are not radial symmetric (the blocks). Taken together, this means that the model is capable of learning to distinguish between individual objects and, at least to some extent, of learning to distinguish between shapes. Another observation that can be done is that many centers of activity are equal for different explorations of the same objects. We were very exact when localizing the objects for the hand to explore, i.e. an object was placed in exactly the same way in different explorations. When we carried out another test in which we were not that careful when locating the object in exactly the same way it did not work out that well. The conclusion from this was that the differences between the matrices generated in the STM Module were extremely small and a very exact location of the objects was needed in order for it to work. This also explains why the center of activation in the SOM Module was often the same in different explorations of the same object.

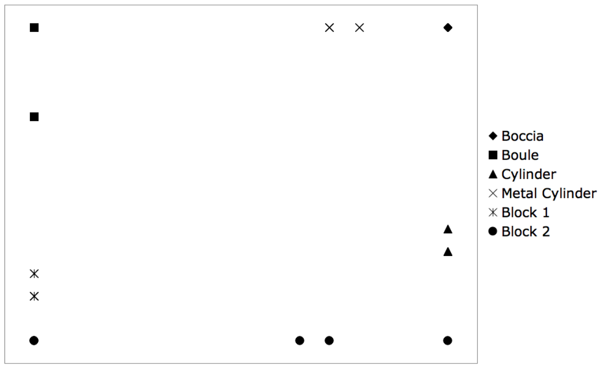

Figure 4: *Results of the grasping experiment with model 1. The depiction shows the centers of activity in the SOM during the testing with the training set patterns.*

3 Model 2

3.1 Design

To overcome the limitation of model 1 which needed the test objects to be accurately localized, we redesigned it into model 2, which uses the tensor product in three steps. Model 2 consists of the Lucs Haptic Hand II, the sensory and motor drivers, the four Ikaros modules used in model 1, and in addition three instances of an Ikaros module dedicated to carrying out the tensor product operation. In model 2 the Grasping Module does not receive input from the motor driver and does not calculate the tensor product. Instead the tensor product calculations are done in the dedicated Ikaros module.

In this model the proprioceptive information is divided into three vectors. One vector represents proprioceptive information about the height of the hand, one vector represents proprioceptive information about the angle of the wrist, and finally one vector represents the configuration of the hand.

One instance of the Tensor Product Module takes as input the part of the proprioceptive output vector from the motor driver that represents the height of the hand and the part that represents the angle of the wrist. The output from this instance of the Tensor Product Module is sent to another instance of the same module. In addition, the second instance of the module takes the vector that represents the configuration of the hand as input. The output from the second instance of the Tensor Product Module is conveyed to a third instance of the same module that also takes the sensory vector from the sensory driver as input. The result of this chain of operations is a successive recoding of the spatial coordinate of the sensor at each joint which makes the final matrix implicitly code the three dimensional location of the sensor when it reacts to the object. This is an important difference from model 1 where different shapes in principle could result in the same coding.

The resulting matrix is sent as output to the STM

Table 1: *The test objects used with the three haptic models for the LUCS Haptic Hand II*

Object	Material	Size (mm)	Size (mm)	Size (mm)
Boccia	Plastic	Diameter = 72	-	-
Boule	Metal	Diameter = 82	-	-
Cylinder	Hard Board	Diameter = 62	Height = 121	-
Metal Cylinder	Metal	Diameter = 75	Height = 109	-
Block 1	Wood	Height = 110	Length = 50	Width = 50
Block 2	Wood	Height = 110	Length = 58	Width = 50

Module. This module integrates the all the sensor readings over the object and forms a code that depends on the three-dimensional shape of the object. As in model 1, the output from the STM Module is used as input to a SOM Module with 225 neurons.

3.2 Grasping Tests

The grasping tests with model 2 were carried out in a similar way as those done with model 1 and the same set of test objects was used. The only differences were that new explorations were done with the model 2, and we were not as careful as before when localizing the test objects in different explorations. To simplify the test procedure the tactile information generated during the explorations of the test objects were written to and read from files. As with model 1 each test object was explored 5 times.

3.3 Results and Discussion

This model was able to discern all the objects and also to categorize them according to shape, i.e. the spheres were categorized in one area of the SOM, the blocks were categorized in another, and the cylinders were categorized in still another area of the SOM (Fig. 6). This might be compared with the discrimination ability of a child younger than 12 months. Such a child is able to discriminate course differences between objects placed in its hand, for example the child is able to discriminate between a cube and a cylinder but not between a cube and a cross-like stimuli (Millar, 2006).

4 Model 3

4.1 T-MPSOM

This model (Fig. 7) is similar to model 2 that uses the tensor product in several steps. The difference is that model 3 has substituted the tensor product operations with instances of a neural network. This neural network (T-MPSOM) is a variant of the SOM that allows multiple peak activations and receives two inputs. It is probably possible to generalize the idea to an arbitrary number of inputs.

There are several advantages when using the T-MPSOM instead of using the tensor product. One advantage is that the model becomes more biologically plausible. Another advantage is that the T-MPSOM is able to downscale the dimensions of the representation, which cannot be done with the tensor product. In model 3 the third instance of T-MPSOM consists of 1058 neurons, while the third instance of the tensor product module outputs a matrix with 270000 elements.

The T-MPSOM consists of a two-dimensional grid of neurons, where every neuron has two weight vectors corresponding to the dimensionality of the respective input vectors.

The sum of each input element multiplied with an arbor function (Dayan, 2000), which corresponds to the receptive field of the neuron, multiplied with the weight is calculated for both inputs. These sums are then multiplied with each other to yield the activity of the neuron.

All neurons in the neural network contribute to the updating of the weight vectors. The level of contribution from a certain neuron depends on the activity in the contributing neuron and a Gaussian function of the distance between the contributing neuron and the neuron with the weight considered for updating. This yields a multiple peak SOM. In every iteration, the input vectors and the weight vectors are normalized.

In mathematical terms, the T-MPSOM consists of a $i \times j$ matrix of neurons. In each iteration every neuron n_{ij} receives the two input vectors $a \in \Re^m$ and $b \in \Re^n$. n_{ij} has two weight vectors ${w_a}^{ij} \in \Re^m$ and ${w_b}^{ij} \in \Re^n$. The activity in the neuron n_{ij} is given by

$$x_{ij} = \sum_m A(i,m){w_a}^{ij}{}_m a_m \sum_n A(j,n){w_b}^{ij}{}_n b_n$$

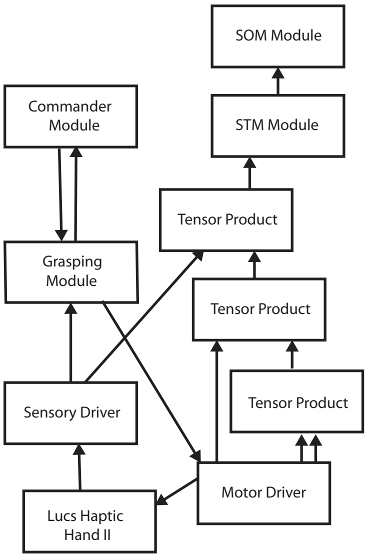

Figure 5: *Schematic depiction of model 2. The Lucs Haptic Hand II conveys the status of the sensors to the Sensory Driver, which in turn conveys the status of the sensors as an array to the Grasping Module and to the uppermost Tensor Product Module. The Motor Driver conveys, to the Lucs Haptic Hand II, the wanted positions of the servos in the robot hand and the wanted time to reach these positions. In addition, the Motor Driver conveys proprioceptive information, i.e. information about the configuration of the robot hand to the second Tensor Product Module. The Motor Driver also conveys to vectors to the lowermost Tensor Product Module. One of these vectors contains information about the wrist angle and the other contains information about the height of the robot hand. The lowermost Tensor Product Module conveys the resulting matrix of the tensor product between its to input vectors. The matrix is transformed into a vector by putting the rows after each other. The second Tensor Product Module conveys its output to the uppermost Tensor Product Module, again transformed to a vector with the matrix rows after each other. The Grasping Module conveys the wanted configuration of the robot hand and the wanted time to reach it to the Motor Driver, the status of individual grasps to the Commander Module. The uppermost Tensor Product Module conveys its output to the STM Module. The STM Module conveys a matrix of the tensor product superimposed from a whole exploration to the SOM Module. The Commander Module conveys orders to initiate grasping movements to the Grasping Module.*

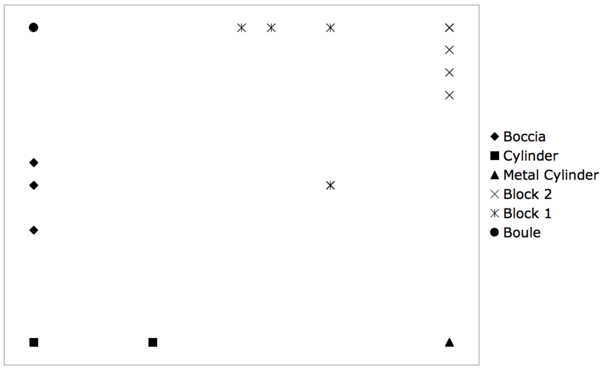

Figure 6: *Results of the grasping experiment with model 2. The depiction shows the centers of activity in the SOM during the testing with the training set patterns.*

in the variant with multiplied activations, and by

$$x_{ij} = \sum_m A(i,m){w_a}^{ij}{}_m a_m + \sum_n A(j,n){w_b}^{ij}{}_n b_n$$

in the variant with added activations, where

$$A(\alpha,\beta) \propto e^{-\left(\alpha - \frac{max\alpha}{max\beta}\beta\right)^2/2\sigma^2}.$$

The updating of the weight vectors are given by

$${w_a}^{ij}(t+1) = {w_a}^{ij}(t) - \alpha(t)\beta_{ij}(t)\left[a(t) - {w_a}^{ij}(t)\right]$$

and

$${w_b}^{ij}(t+1) = {w_b}^{ij}(t) - \alpha(t)\beta_{ij}(t)\left[b(t) - {w_b}^{ij}(t)\right],$$

where $0 \leq \alpha(t) \leq 1$, and $\alpha(t) \to 0$ when $t \to \infty$. The learning in each neuron is controlled by

$$\beta_{ij}(t) = \frac{\beta'_{ij}(t)}{max\beta'_{ij}(t)}$$

where

$$\beta'_{ij}(t) = \sum_k \sum_l x_{kl}(t)G(n_{kl}, n_{ij})$$

and $x_{kl}(t)$ is the activity in n_{kl} at time t and $G(n_{kl}, n_{ij})$ is a Gaussian function.

4.2 Grasping Tests

To simplify the test procedure with the model, the tactile information generated during the explorations of the objects in Table 1 were written to files. As before, 5 explorations were carried out with each object. The

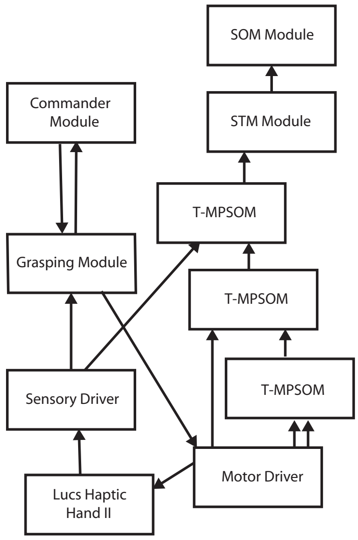

Figure 7: *Schematic depiction of model 3. The difference between this model and model 2 is that the Tensor Product Modules are replaced with T-MPSOM neural networks.*

model was first trained with randomly chosen explorations from the training set during, in total, 5000 iterations (one exploration consists of approximately 270 iterations). This was done to let the three instances of the T-MPSOM self-organize. When the learning had converged the model was exposed to each sample of the training set, i.e. in total 30 explorations and the output matrices from the STM Module were written to files. This set of 30 files was then used as a training and test set for the SOM Module. The model was trained with 1000 randomly chosen samples from its training set, after which the trained model was tested with all 30 samples in the training set.

4.3 Results and Discussion

The result from the grasping tests with this model (Fig. 8) is that the blocks, the cylinders and the spheres were categorized in different areas of the SOM. The model also separates the individual objects in the training/test set with only one exception when a boccia sphere was taken as a boule sphere. Therefore the capacity of this model is comparable to model 2 described above. We also experimented by varying the relationship between the input vectors and the size of the T-MPSOM in x and y directions. We found that the performance was superior when the relationship between the number of elements in input vector a and the number of neurons in the y direction

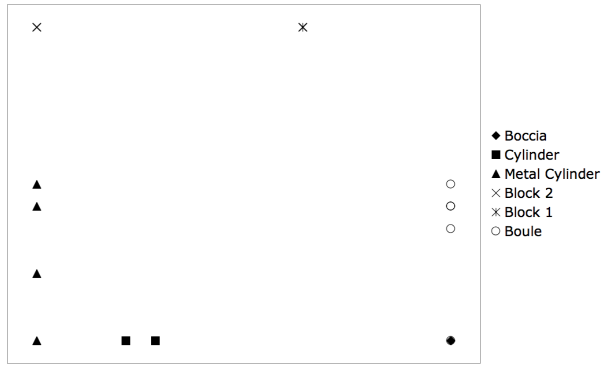

Figure 8: *Results of the grasping experiment with model 3. The figure shows the centers of activity in the SOM during the testing with the training set patterns.*

was similar to the relationship between the number of elements in input vector b and the number of neurons in x direction. This is probably due to that the activity is otherwise smeared out. Further we found that the sigma of the arbor function should be set rather small for the model to work well. The three instances of the T-MPSOM used 25, 90 and 1058 neurons. The SOM consisted of 225 neurons as in models 1 and 2.

5 Conclusions

We have designed and implemented three haptic models together with the Lucs Haptic Hand II. All three of them did, more or less, manage to categorize the test objects according to shape. Model 2, the one that used the tensor product in several steps, and model 3 with the novel T-MPSOM network worked best. These models are capable of both the categorization of the test objects according to shape, and to identify individual objects, with one exception for model 3, i.e. it once categorized a boccia sphere as a boule sphere. If model 3 had used a greater number of neurons it would probably not have been mistaking in any case. However, we avoided simulations with more neurons because we wanted a model that was not too computationally heavy, and the capacity of the current model appears to be comparable to that of a human. If the model were implemented in hardware a larger number of neurons would be acceptable. The worst performing model was model 1. That model did not perform very well in the categorization according to shape, and in addition it required a great accuracy in the location of a test object.

In the future we will further investigate the potential of T-MPSOM based models. In exchanging the STM and SOM modules to a SOM module with leaky integrator neurons we hope to obtain a haptic system consisting of artificial neural networks that self-organize in accordance with the input in all its parts. We will also experiment with a variation of the T-MPSOM that instead of multiplying add the activity contributions from each of the input vectors. Later we will study the interaction between haptics and vision.

Acknowledgements

We want to express our acknowledgements to Stiftelsen Elisabeth Rausings Minnesfond for finacial support for the Lucs Haptic Hand II, to Lars Kopp for the lifting mechanism of the robot hand and to Jacek Malec for helpful comments on the manuscript.

References

Arbib, M. A., Billard, A., Iacoboni, M., & Oztop, E. (2000). Synthetic brain imaging: grasping, mirror neurons and imitation. *Neural Networks*, 13, 975-999.

Balkenius, C., and Morén, J. (2003). From isolated components to cognitive systems. *ERCIM News, April 2003, 16*.

Coelho, J., Piater, J., & Grupen, R. (2001). Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot, *Robotics and Autonomous Systems*, 37, 2-3, 195218.

Dario, P., Laschi, C., Carrozza, M.C., Guglielmelli, E., Teti, G., Massa, B., Zecca, M., Taddeucci, D., & Leoni, F. (2000). An integrated approach for the design and development of a grasping and manipulation system in humanoid robotics, *Proceedings of the 2000 IEEE/RSJ international conference on intelligent robots and systems*, 1, 1-7.

Dario, P., Laschi, C., Menciassi, A., Guglielmelli, E., Carrozza, M.C., & Micera, S. (2003). Interfacing neural and artificial systems: from neuroengineering to neurorobotics, *Proceedings or the 1st international IEEE EMBS conference on neural engineering*, 418-421.

Dayan, P. (2000). Competition and Arbors in Ocular Dominance, *NIPS 2000*, 203-209.

DeLaurentis, K.J., & Mavroidis, C. (2000). Development of a shape memory alloy actuated robotic hand. (2004-10-28). http://citeseer.ist.psu.edu/383951.html

Fagg, A. H., & Arbib, M. A. (1998). Modeling parietal premotor interactions in primate control of grasping. *Neural Networks*, 11 (7 8), 1277-1303.

Heidemann, G., & Schöpfer, M. (2004). Dynamic tactile sensing for object identification, *Proceedings. ICRA '04. 2004 IEEE International Conference on Robotics and Automation*, 2004, 1, 813-818.

Hosoda, K., Tada, Y., & Asada, M. (2006). Anthropomorphic robotic soft fingertip with randomly distributed receptors, *Robotics and Autonomous Systems*, 54, 2, 104-109.

Jockusch, J., Walter, J., & Ritter, H. (1997). A tactile sensor system for a three-fingered robot manipulator, *Proceedings, 1997 IEEE International Conference on Robotics and Automation*, 1997, 4, 3080-3086.

Johnsson, M. (2004). Lucs Haptic Hand I - Technical Report, *Lucs Minor*, 8.

Johnsson, M. (2005). http://www.lucs.lu.se/People/Magnus.Johnsson/HapticPerception.html

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.

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

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

Johnsson, M., & Balkenius, C. (2006b). Lucs Haptic Hand II, *Lucs Minor*, 9.

Kohonen, T. (2001). *Self-organizing maps*, Berlin, Springer-verlag.

Millar, S. (2006). Network models for haptic perception, *Infant Behavior and Development*, 28, 3, 250-265.

Rhee, C., Chung, W., Kim, M., Shim, Y., & Lee, H. (2004). Door opening control using the multi-fingered robotic hand for the indoor service robot, *Proceedings of the 2004 IEEE international conference on robotics & automation, 4*, 4011-4016.

Sugiuchi, H., Hasegawa, Y., Watanabe, S., & Nomoto, M. (2000). A control system for multi-fingered robotic hand with distributed touch sensor, Industrial electronics society. *IECON 2000. 26th annual conference of the IEEE, 1*, 434-439.