GBF NETWORK ARCHITECTURES FOR ROBOT VISION

JOSEF PAULI
Christian–Albrechts–Universität, Institut für Informatik,
Preusserstrasse 1-9, D–24105 Kiel, Germany

ABSTRACT:
A versatile robot manipulator is based on techniques of computer vision and neural network learning. For grasping objects four principal tasks have to be done in a cycle — detect the desired object and the grasping fingers in the images, evaluate the spatial relationship wrt. grasping stability, choose a more stable grasping pose (if possible), and move the manipulator to it. In this work we focus on two subproblems thereof, specifying the hand–eye coordination and evaluating the grasping situation. These procedures are based on (nonlinear) functions which are not known a priori and therefore have to be learned. We uniformly approximate the required functions by means of networks of Gaussian Basis Functions (GBF networks). Modifying the number of GBFs and/or the accessory size of the gaussian support changes the quality of the learned function. We use GBF networks both to specify the hand–eye coordination and to recognize the grasping situation and show how various network configurations influence the accuracy and therefore the usefulness of the function approximations. All experiments are carried out in real world environments using an industrial robot manipulator and the computer vision system KHOROS.

INTRODUCTION

The motto of William of Occam, a scholastic of the middle ages, reads as follows: It's vain to do with more what can be done with less. This principle, known as Occam's Razor (Blumer et al., 1990), is very much alive in neural network learning. Several theoretically oriented papers study the aspects of learnability and derive lower and upper bounds on the sample size versus net size needed such that a function approximation of a certain quality can be expected (Baum and Haussler, 1990). A sample is a set of input–output pairs (examples) characterizing the desired function. Most recently, the relationship between sample compression and probably approximately correct learning (PAC–learnability) has been discussed (Floyd and Warmuth, 1995). The term PAC–learnability characterizes a function as learnable if and only if a learning algorithm can be formulated which produces with a certain expenditure of work a probably approximately correct function. Based on pre–specified levels for probability and correctness of a function approximation, we are interested in discovering the

most simply and efficiently applicable representation (Rissanen, 1984).

An approximation scheme leading to a function which is close to the sample data and meets smoothness constraints is known under the term regularization. It was shown that regularization principles can be implemented with neural networks consisting of an input layer, a layer of hidden nodes and an output layer (Poggio and Girosi, 1990). Each hidden node is defined by a so–called support basis function acting as a window in the input space. In our application we use GBF networks whose basis functions are multidimensional symmetric Gaussians. On account of applying Occam's Razor to GBF networks it is desirable to discover the minimum number of GBFs to reach a critical quality for the function approximation. Our work treats that problem from a practical point of view by doing real world experiments in vision based robotics. We show the relationship between net size and/or support size of a GBF network on the one hand and the quality of the function approximation on the other hand. Various configurations of GBF networks are applied for eye–hand coordination and for evaluation of grasping situations. Taking the experimental results into account we can configure appropriate GBF networks for vision based robot grasping.

EYE–HAND COORDINATION

For grasping an object the end–effector of the robot manipulator has to be moved into a stable grasping pose. The desired pose (position and orientation) must be extracted from visual information which will be produced by two cameras. The camera system is arranged in an appropriate position and orientation for watching the scene (no physical connection to the robot). Taking stereo images and detecting the target object in the two images result in two twodimensional positions representing the centers of gravity (two 2D–vectors). The two positions are defined in the coordinate systems of the two cameras and will be combined in a single vector (4D–vector). On the other hand, the end–effector moves within a 3D working space which is defined in the basis coordinate system of the robot (the position of the end–effector is a 3D–vector). Thus we need a function transforming the object positions from the coordinate systems of the cameras to the cartesian coordinate system of the robot (4D–vector $\Longrightarrow$ 3D–vector).

The procedure to acquire that function, which determines the eye–hand coordination, is as follows. We make use of a training sample for learning a GBF network. First the set of GBFs must be configured, and second the combination factors of the GBFs are computed. We configure the set of GBFs by simply selecting certain elements from the training sample and using the input parts (4D–vectors) of the selected examples to define the centers (of the GBFs). The combination factors for GBFs are computed with the pseudo inverse technique, leading to least square errors between pre-specified and computed output values.

The prerequisite for running the learning procedure is the existence of a training sample. To get it, we take full advantage of the robot dexterity. The end–effector moves in the working space systematically, stops on equidistant places, and 3D–positions of the end–effector are carefully recorded. These 3D-vectors are simply provided by the control unit of the robot. Furthermore, at

each stopping place an SSD–based (sum of squared distances) recognition algorithm detects the end–effector bend in the stereo images (see Figure 1) and the two twodimensional positions are combined to a 4D-vector. Alternative striking features, e.g. the end–effector tip, could be detected as well. All pairs of 4D–/3D–vectors are used as training sample for the desired eye–hand coordination.

Based on image coordinates of the end–effector bend the GBF network has to estimate its 3D position in the robot basis coordinate system. The mean 3D position error should be as low as possible. The main question of interest is: How many GBFs and which support sizes are needed to get a certain quality for the eye–hand coordination? To answer that question four experiments have been carried out. In the first and second experiment we applied two different numbers of GBFs exemplarily. The third experiment shows the effect of doubling the image resolution. Finally, the fourth experiment takes special care for training the combination weights of the GBFs. In all four experiments we systematical increase the GBF support size and measure the mean position error.

For each experiment we take a training sample. The working space of the end–effector underlying the sample is cube–shaped of maximum 300 millimeters (mm) side length. The GBFs will be spread over a subspace of 4D–vectors according to certain stopping places of the end–effector. That is, the 4D image coordinates resulting from the end–effector bend position at a certain stopping place are used for defining the center of a Gaussian. The following experiments differ wrt. the size and the use of the training sample. The evaluation of the resulting GBF network is based on a testing sample. It consists of input–output pairs from the same working space as above, but definitely the robot fingers moved in discrete steps of 20 mm. It is assured, that training and testing sample differ essentially and have only a small number of elements in common.

In the first experiment the manipulator moved in discrete steps of 50 mm through the working space resulting in a training sample of 7x7x7 = 343 elements. Every second example is used for defining a GBF (4x4x4 = 64 GBFs), and the whole training sample for computing the combination weights of the GBFs. The image resolution is set to 256x256 pixel. Figure 2 shows in curve (a) the course of mean position error for systematically increasing the support. As the GBFs become more and more overlapped the function approximation improves, and the mean position error decreases to a value of about 2.2 mm.

The second experiment differs from the first in that the manipulator moved in steps of 25 mm. Thus the training sample consists of 13x13x13 = 2197 examples and every second example is used for defining a GBF (7x7x7 = 343 GBFs). Figure 2 shows in curve (b) that the mean position error converges to 1.3 mm.

In the third experiment the same configuration was used as before, but the image resolution is doubled to 512x512 pixels. The accuracy of detecting the finger bend in the images increases, and the mean position error decreases further. Figure 2 shows in curve (c) the convergence to error value 1.0 mm.

The fourth experiment takes special care of both the training of weights and the testing of the resulting GBF network. It is obvious, that at the border of the working space there is only a one–sided overlap between GBFs. Therefore the quality of the function approximation can be improved, if a specific subset

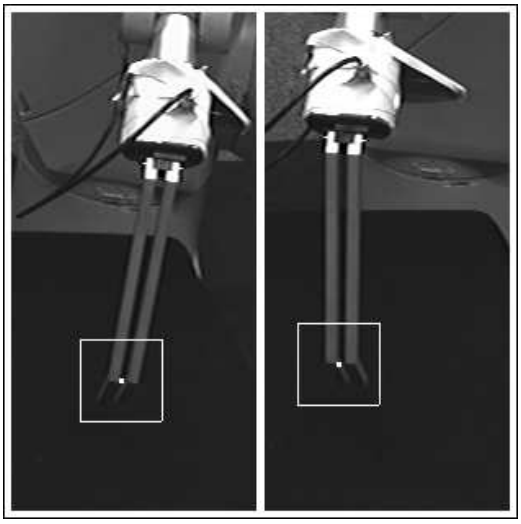

Figure 1: Detecting end–effector in stereo images.

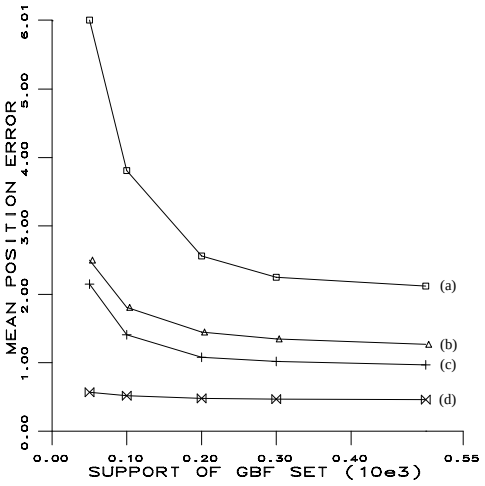

Figure 2: Courses of mean position error.

of 4D–/3D–vectors located at the border of the working space will not be taken into account. In this experiment, the 343 GBFs are spread over the original working space as before, but for computing combination factors and testing the GBF network an inner working space of 250 side length is used. Figure 2 shows in curve (d) that the mean position error decreases to a value of 0.5 mm.

We used GBF networks to learn the mapping from image coordinates in stereo images into coordinates of a robot manipulator. The main advantage is that extrinsic and intrinsic camera parameters don't have to be determined. Rather we configure a GBF network appropriately, depending on the required accuracy (e.g. 1 mm). In order to approach the manipulator to a target object, we detect the target in stereo images, compute the centers of gravity, put the image coordinates into the GBF network, and compute the relevant robot coordinates.

EVALUATION OF GRASPING SITUATION

Having the manipulator near the object we must fine-tune the pose of the robot fingers in order to stably grasp the target. Therefore the spatial arrangement between target and fingers have to be recognized and evaluated wrt. grasping stability. According to our vision based approach the image depiction of that arrangement is used to draw conlusions about the grasping stability. Figure 3 shows three images each depicting a target object, two bended robot fingers, and some other objects. On the left and the right the grasping situation is unstable because the horizontal part of the two parallel fingers is behind respectively in front of the target. The grasping situation in the middle image is most stable.

The grasping situations will be evaluated by applying a specific function to the contents of the relevant image area. To learn that evaluation function a GBF network is used. The input nodes describe the relevant image area, the hidden nodes represent a set of typical grasping situations, and the only output node computes the grasping stability. There are many approaches known in the computer vision literature for describing image contents. In this application we prefer

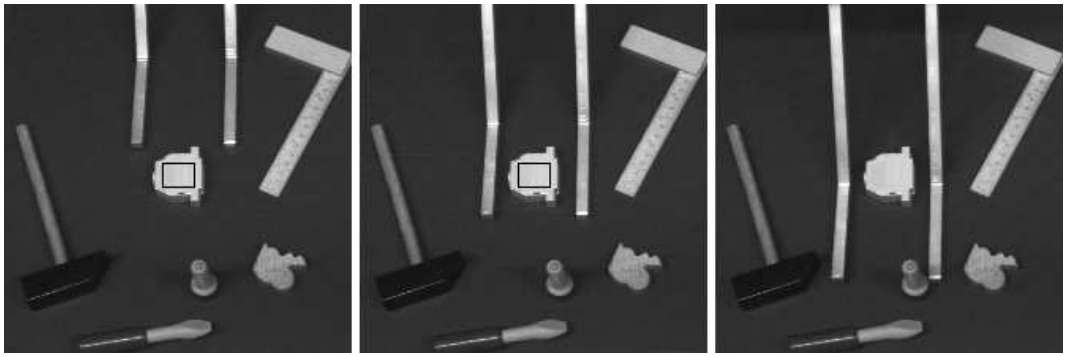

Figure 3: Three typical arrangements of target object and grasping fingers.

Figure 4: Response of Gabor filter.

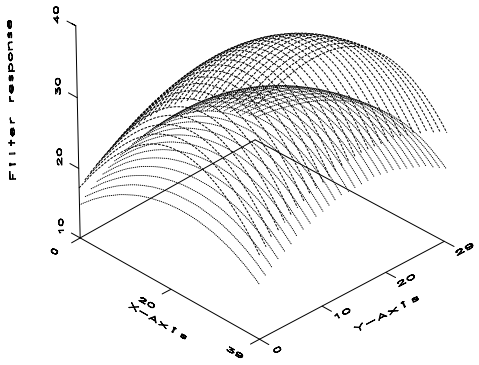

Figure 5: Overlay of two filter response patterns.

to use the response of a simple filter operation and avoid image segmentation. Thus we don't have to bridge the problematic gap between photometric gray level edges and geometric surface discontinuities (Maxwell and Shafer, 1994).

In line with this concept it would (!) be possible to take the raw gray levels and use directly the appearance patterns of grasping situations. Unfortunately these patterns are large–sized and so the efficiency of recognition is low. Therefore, we are interested in an image operator which concentrates the contents of a large image area into a smaller patch. The Gabor wavelet filter can be applied for this purpose (Pauli et al. 1995). For example, Figure 4 shows the response of an adequately parameterized Gabor filter applied to the left image in Figure 3. Rather than using appearance patterns we take so-called response patterns of a pre–defined size for evaluating the grasping situation. Specific arrangements consisting of the fingers and the target result in specific filter response patterns. For example, Figure 5 shows the overlay of two response patterns which result by applying the Gabor filter to the left and the middle image in Figure 3 and selecting the response of the (black) outlined rectangular area.

Unlike the simple application of GBF networks in eye–hand coordination, the dimension of the evaluation function (for grasping situations) is extremely high. Definitely, the dimension of the input space is equal to the number of pixels of the response patterns (the pre–defined size typically has several hundred pixels),

and furthermore the GBFs have to be defined according to this dimension. The input space of the GBF network is the set of all possible response patterns of the pre–defined size. Each hidden node only responds significant for a certain subset of these patterns. The factors combining the GBFs encode values of grasping stability assigned to typical grasping situations. The output node computes the definite grasping stability for the grasping situation put into the input layer.

The approach for learning the evaluation function is as follows:
First, we take example images containing various grasping situations. Especially in our experiment the robot fingers will be moved step by step to the most stable grasping situation and step by step moved off afterwards. The movement is photographed in 25 discrete steps (Figure 3 shows three images thereof). Every second image will be used for training and the images in between for testing.
Second, we apply the Gabor filter to the training images and extract the rectangular area (of pre-specified size) describing the grasping situation. This training sample of response patterns is used for learning the evaluation function.
Third, according to the approach for learning a GBF network we have to cluster the response patterns with regard to similarity. The ISODATA clustering procedure is used for this purpose (Schalkoff, 1992, pp. 109-125).
Finally, we determine appropriate combination factors of the GBFs using the pseudo inverse technique. For that a set of pre-specified stability values must be assigned to the training sample. Considering the order in which the examples of grasping situations have been photographed, we define that the course of stability values should take the form of a bell–shaped parabolic curve. Therefore, the course of stability value for the ordered set of training examples increases systematically until the maximum is reached and decreases afterwards.

According to this approach a GBF network can be configured representing an evaluation function. Four experiments have been carried out by taking different numbers and/or support sizes of the GBFs. Figure 6 shows in curve (a) and (b) the course of stability values if we take six GBFs and a small respectively large support size. Alternatively, the curves (c) and (d) in Figure 7 depict the courses for 13 GBFs and a small respectively large support size. Curve (d) depicts the best approximation of the evaluation function.

Using such a GBF network for evaluating the grasping situation, the robot system automatically controls the manipulator in order to reach the optimal grasping stability.

CONCLUSION

Our approach of vision based robotics uses GBF networks both for eye–hand coordination and for the evaluation of grasping situations. Furthermore, GBF networks can be used to learn operators for view independent object recognition (Pauli, 1996). In numerous experiments it was demonstrated how specific network configurations influence the accuracy of the function approximation. Depending on pre-specified limits for the accuracy the GBF networks can be trained appropriately and then used for online operation.

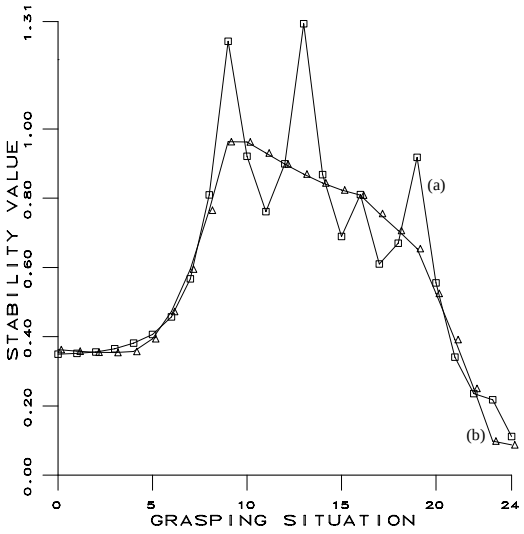

Figure 6: Courses of grasping stability.

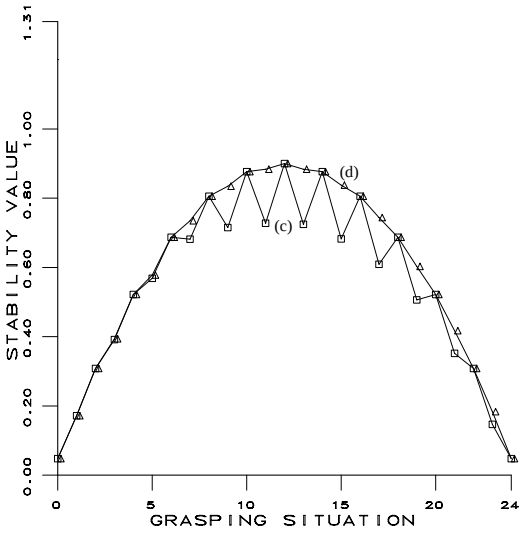

Figure 7: Courses of grasping stability.

REFERENCES

Baum E., Haussler D., (1990). What net size gives valid generalization?, in Shavlik J., Dietterich T. (1990), Readings in Machine Learning, Morgan Kaufmann Publishers, 258-262.

Blumer A., Ehrenfeucht A., Haussler D. (1990). Occams' Razor, in Shavlik J., Dietterich T. (1990), see above, 201-204.

Floyd S., Warmuth M., (1995). Sample compression, learnability, Vapnik–Chervonenkis Dimension, Machine Learning Journal, Vol. 21(3), 269-304.

Maxwell B., Shafer S., (1994). A framework for segmentation using physical models of image formation, IEEE Conference on Computer Vision and Pattern Recognition, 361-368.

Pauli J., Benkwitz M., Sommer G., (1995). RBF networks for object recognition, in Krieg-Brückner B., Herwig Ch. (1990), Workshop Kognitive Robotik, Zentrum für Kognitionswissenschaften, Universität Bremen, Bericht 3/95.

Pauli J., (1996). View independent object recognition, British Machine Vision Conference, Edinburgh.

Poggio T., Girosi F., (1990). Networks for approximation and learning, Proceedings of the IEEE, Vol. 78, 1481-1497.

Rissanen J., (1984). Universal coding, information, prediction, and estimation, IEEE Transactions on Information Theory, Vol. 30(4), 629-636.

Schalkoff R. (1992). Pattern Recognition - Statistical, Structural, and Neural Approaches, John Wiley and Sons, New York.