, , 1–21 ()
© Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Learning to Recognize and Grasp Objects

JOSEF PAULI jpa@informatik.uni-kiel.de
Christian–Albrechts–Universität, Institut für Informatik, Preusserstrasse 1-9, D–24105 Kiel, Germany

Editor: Henry Hexmoor and Maja Mataric

Abstract. We apply techniques of computer vision and neural network learning to get a versatile robot manipulator. All work conducted follows the principle of autonomous learning from visual demonstration. The user must demonstrate the relevant objects, situations, and/or actions, and the robot vision system must learn from those. For approaching and grasping technical objects three principal tasks have to be done — calibrating the camera–robot coordination, detecting the desired object in the images, and choosing a stable grasping pose. These procedures are based on (nonlinear) functions, which are not known a priori and therefore have to be learned. We uniformly approximate the necessary functions by networks of gaussian basis functions (GBF networks). By modifying the number of basis functions and/or the size of the gaussian support the quality of the function approximation changes. The appropriate configuration is learned in the training phase and applied during the operation phase. All experiments are carried out in real world applications using an industrial articulation robot manipulator and the computer vision system KHOROS.

Keywords: Radial basis function networks, learning from visual demonstration, camera–robot calibration, approaching and grasping objects, recognition of objects, recognition of geometric relations.

1. Introduction

We have equipped a robot manipulator with a vision system for autonomous grasping and assembling technical objects. The vision system deals with two sub–tasks, recognizing the target object in the image and evaluating the stability of grasping situations. The transformation of the image coordinates of a target object into 3D coordinates for the robot is based on the representation of the camera–robot coordination.

The robot vision system must recover those partial scene information from the image, which is indispensable needed to recognize and manipulate an object (see principles of purposive vision in (Aloimonos, 1993)). For example, object recognition has to be grounded on features, which discriminate between the target object and other objects and can be extracted from the image easily. Furthermore, for appropriate grasping a target object, we are interested in features, which describe or evaluate the situation under the criterion of grasping stability (Cutkosky, 1989).

We have implemented a vision system, which is constructional in the sense, that operators for recognizing target objects can be learned for the actual environment. The operators are based on $2D$ *appearance patterns* of the relevant objects or *response patterns* resulting from specific filter operations. The most closely related work for object recognition is from (Murase & Nayar, 1995), who describe a method for *appearance based object recognition*. An appearance manifold of the target object must be acquired by changing the object view angle systematically and taking images in discrete steps. Based on the most relevant eigenvectors of the covariance matrix the *Karhunen–Loeve transform* is used for compressing the manifold. As a linear reduction of the dimension takes place, the approach is only suitable when the data are approximate linear distributed. Alternatively, we show the use of networks of gaussian basis functions (GBF networks) for appearance based object recognition. The approach allows a *nonlinear dimension reduction* and does not assume linear constrained appearance manifolds. By carefully spreading and configuring basis functions, an optimal operator can be learned, which carries out a compromise between the invariance and reliability criterion.

Our approach can be used as well for recognizing and evaluating grasping situations. Based on visual information the manipulator will be servoed to the most stable grasping pose in order to grasp the target object finally.[1] Similar to (Kamon, Flash, & Edelman, 1994), we use visual information for grasping, but in opposition to their work, we avoid to extract geometric data explicitely and instead leave the geometric data implicit in the appearance manifold of grasping situations. By refraining from image segmentation, there is no need to bridge the problematic gap between photometric gray level edges and geometric surface discontinuities (Maxwell & Shafer, 1994). For grasping objects the $3D$ geometric shape is not required in full detail (exemplary pointed out in (Trobina, Leonardis, & Ade, 1994)), and according to that, we only extract the necessary information for evaluating the grasping stability.

We not only use GBF networks for recognizing target objects and grasping situations but also for learning the camera–robot coordination. The neural network implicit represents three blocks of information: the intrinsic camera properties, the geometric relation between camera and robot, and the attributes for reconstructing the position of an object from stereo image coordinates into robot coordinates. Usually, the relation between $3D$ space coordinates and $2D$ image coordinates is written linear by projective transformation (Faugeras, 1993). However, this is not acceptable for certain types of camera objectives (e.g., lenses of small focal length). By spreading more gaussian basis functions into critical areas of the input space, we can better approximate the input–output mapping and hence take care for nonlinearities in the projective transformation.

On account of applying the principle of minimum description length (Rissanen, 1984) to configuring the neural networks, it is desirable to discover the minimum number of basis functions for a certain critical quality of the function approximation. Our work treats that problem from a practical point of view by doing real world experiments with the robot vision system. We show the relationship between net size or support size on the one hand and the *overfitting versus overgeneralizing conflict* on the other hand.

Section 2 introduces the regularization principle for function approximation and derives from that the definition of GBF networks. Furthermore, the approach for learning the basis functions and accompanying combination factors is described. Section 3 illustrates the procedure for camera–robot coordination and explains how to apply GBF networks for learning and representing that relationship. Section 4 gives an overview to the approach of learning image operators, which are needed to recognize target objects or grasping situations. Sections 5 and 6 present experiments on learning image operators for object recognition under varying viewing conditions. This is important for typical real world applications of object grasping. In section 7, the system is applied to learning to recognize grasping situations (geometric relation between the robot fingers and a target object) with the intention of evaluating the grasping stability. Section 8 summarizes and discusses the work.

2. Regularization principles and GBF networks

An approach for function approximation is needed, which has be grounded on sample data of the input–output relation. The function approximation should fit the sample data to meet *closeness constraints* and should generalize over the sample data to meet *smoothness constraints*. Neglecting the aspect of generalizing leads to *overfitted functions*, otherwise, neglecting the fitting aspect leads to *overgeneralized functions*. Hence, both aspects have to be combined to get a qualified function approximation. The *regularization approach* incorporates both constraints and determines such a function by minimizing a functional (Poggio & Girosi, 1990).

Let $S := \{(\vec{p_i}, r_i) | (\vec{p_i}, r_i) \in (R^m \times R); i = 1, \cdots, N\}$ be the sample data representing the input–output relation of the function that we want to approximate. The functional in equation (1) consists of a *closeness term* and a *smoothness term*, which are combined by a factor λ expressing relative importance of each.

$$H(f) := (\sum_{i=1}^{N} (r_i - f(\vec{p_i}))^2) + \lambda \parallel P(f) \parallel^2 \tag{1}$$

The first term computes the sum of squared distances between the desired and the actual outcome of the function. The second term incorporates a differential operator P for representing the smoothness of the function.

Under some pragmatic conditions (see again (Poggio & Girosi, 1990)) the solution of the regularization functional is given by equation (2).

$$f(\vec{p}) := \sum_{i=1}^{N} v_i G_i(\vec{p}, \vec{p_i}) \tag{2}$$

The basis functions G_i are *gaussians*, specified for a limited range of definition, and having $\vec{p_i}$ as the centers. Based on the non–shifted *gaussian support function* G, we get the N versions G_i by shifting the center of definition through the input space to the places $\vec{p_1}, \cdots, \vec{p_N}$. The solution of the regularization problem is a linear combination of *gaussian basis functions* (see equation (2)).

The number of GBFs must not be equal to the number of samples in S. It is of interest to discover the minimum number of GBFs, which are needed to reach a critical quality for the function approximation. Instead of using the vectors $\vec{p_1}, \cdots, \vec{p_N}$ for defining GBFs, we cluster them into M sets (with $M \leq N$) striving simultaneous for minimizing the variances within and maximizing the distances between the sets. From each set a mean vector $\vec{c_i}, i \in \{1, \cdots, M\}$, is selected (or computed). A procedure similar to the error–based *ISODATA clustering algorithm* in (Schalkoff, 1992, pp. 109–125) is used. Initially, the algorithm groups the vectors by using the standard *K–means* method. Then, clusters exhibiting large variances are split in two, and clusters that are too close together are merged. Next, K–means is reiterated taking the new clusters into account. This sequence is repeated until no more clusters are split or merged.

Each typical vector $\vec{c_j}$ specifies the center of the definition range of a GBF.

$$G_j(\vec{p}, \vec{c_j}) := exp(-\frac{\parallel \vec{p} - \vec{c_j} \parallel^2}{2\sigma_j^2}) \tag{3}$$

The function G_j computes a similarity value between the vector $\vec{c_j}$ and a new vector $\vec{p}$. The similarity is affected by the pre–specified parameter σ_j, which determines the *support size* and shape of the GBF. It is intuitive clear, that the ranges of definition of the functions G_j must overlap to a certain degree in order to approximate the recognition function appropriately. The overlap between the GBFs is just determined by the parameters σ_j. The linear combination of GBFs (reduced set) is defined by the factors w_j.

$$\tilde{f}(\vec{p}) := \sum_{j=1}^{M} w_j G_j(\vec{p}, \vec{c_j}) \tag{4}$$

The approach for determining appropriate combination factors is as follows. First, the M basis functions will be applied to the N vectors $\vec{p_i}$ of the training set. This results in a matrix A of similarity values with N rows and M columns. Second, we define an N-dimensional vector $\vec{h}$ comprising the desired output values for the N training vectors. Third, we define a vector $\vec{w}$, which comprises the unknown combination factors $w_1, \cdots, w_M$ of the basis functions. Finally, the problem is to solve the equation $A\vec{w} = \vec{h}$ for the vector $\vec{w}$. According to (Press, Teukolsky, & Vetterling, 1992, pp. 671–675) we compute the pseudo inverse of A and determine the vector $\vec{w}$ directly.

$$A^{\dagger} := (A^T A)^{-1} A^T, \qquad \vec{w} := A^{\dagger}\vec{h} \tag{5}$$

As opposed to the particular specification of the sample data S, an alternative input–output relation could be defined such that the output part is itself a vector instead of a scalar. In that case, we simply compute a specific set of combination factors of the GBFs for each dimension respectively.

In summary, equations (3) and (4) define an approximation scheme, which can be used for relevant functions of camera–robot calibration, object recognition and situation recognition. The approximation scheme is popular in the neural network literature under the term *regularization neural network* (Bishop, 1995, pp. 164–191), and we will call it *GBF network* to emphasize the gaussians. A GBF network consists of an input layer, a layer of hidden nodes and an output layer. The input layer and output layer represent the input and output of the function approximation, the nodes of the hidden layer are assigned to the GBFs.

We will realize in the applications of the next sections, that GBF network learning is a method, which helps to overcome the serious *bias problem* in high-level machine learning (Utgoff, 1986) and parameter estimation (Press, Teukolsky, & Vetterling, 1992). Actually, it is the biological inspired dynamic structure of the network to be changed and controlled on the basis of error feedback (Bruske & Sommer, 1995), which lets the learning method go beyond pure function approximation.

3. Camera–robot 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 put up in an appropriate position and orientation for observing the scene (no physical connection to the robot). By taking stereo images and detecting the target object in the two images, we obtain two two-dimensional 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). Hence, we need a function for transforming the object positions from the coordinate systems of the cameras to the cartesian coordinate system of the robot ($4D$ vector $\Longrightarrow$ $3D$ vector).

Traditionally, this function is based on principles of stereo triangulation by taking intrinsic parameters (of the camera) and extrinsic parameters (describing the camera–robot relation) into account (Faugeras, 1993). Complicated equation systems would have to be solved to compute these parameters, and probably the resulting parameter values would be inaccurate due to error propagation. As opposed to that, we use GBF networks to learn the mapping from stereo image coordinates into coordinates of a robot manipulator. There are two good reasons for this approach. First, the intrinsic and extrinsic parameters will not be computed explicitely, because the coordinate mapping from stereo images to the robot manipulator is determined in a direct way without intermediate results. Second, by varying number and parametrization of the gaussian basis functions during the training phase, the accuracy of the function approximation can be improved as desired.

The procedure for determining the camera–robot coordination is as follows. We make use of training samples for learning a GBF network. First, the set of GBFs is 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 samples and using the input parts ($4D$ vectors) of the selected samples to define the centers of the GBFs. The combination factors for the GBFs are computed with the pseudo inverse technique, which results in least square errors between pre-specified and computed output values.

The prerequisite for running the learning procedure is the existence of training samples. To obtain them, we take full advantage of the robot agility. 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 supplied by the control unit of the robot. Furthermore, at each stopping place a *SSD–based* (sum of squared distances) recognition algorithm detects the end-effector bend in the stereo images (see Figure 1) and the two two-dimensional positions are combined to a $4D$ vector.[2] All pairs of $4D$–$3D$ vectors are used as training samples for the desired camera–robot 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 obtain a certain quality for the camera–robot 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 systematic increase the GBF support size and evaluate the mean position error.

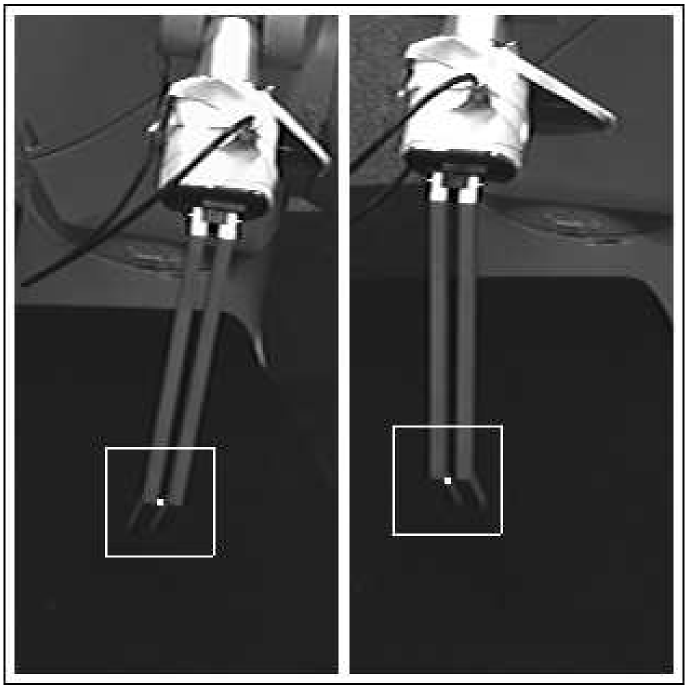

Figure 1. The stereo images show the robot hand with parallel jaw fingers. A SSD-based recognition algorithm has been used to localize the finger bends. This is illustrated by a white square including a central dot.

We take training samples for each experiment. The working space of the end-effector (underlying the samples) is cube–shaped of maximum $300mm$ (millimeters) side length. The GBFs will be spread over a subspace of $4D$ vectors in correspondence to certain stopping places of the end-effector. That is, the $4D$ image coordinates (resulting from the position of the end-effector bend at a certain stopping place) are used for defining the center of a gaussian basis function. The following experiments differ with regard to the size and the usage of the training samples. The application of the resulting GBF networks is based on testing samples. They consist of input–output pairs from the same working space as above, where the robot fingers moves in discrete steps of $20mm$. It is assured, that training and testing samples differ for the most part, i.e., have only a small number of elements in common.

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

The second experiment differs from the first in that the manipulator moved in steps of $25mm$, i.e., $13 \times 13 \times 13 = 2197$ training samples. All samples are used for computing the GBF weights, and every second sample for defining a GBF ($7 \times 7 \times 7 = 343$ GBFs). Figure 2 shows in curve (b), that the mean position error converges to $1.3mm$.

In the third experiment the same configuration has been used as before, but the image resolution was doubled to 512×512 pixels. The accuracy of detecting the finger bend in the images increases, and hence the mean position error of the end-effector bend reduces once again. Figure 2 shows in curve (c) the convergence to error value $1.0mm$.

The fourth experiment takes special care of both the training of weights and the testing of the resulting GBF network. Obviously, there is only a one-sided overlap between GBFs at the border of the working space. Hence, the quality of the function approximation can be improved, if a specific subset of $3D$ vectors, which is 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 an inner working space of $250mm$ side length is used for computing combination factors and for testing the GBF network. Figure 2 shows in curve (d) that the mean position error decreases to a value of $0.5mm$.

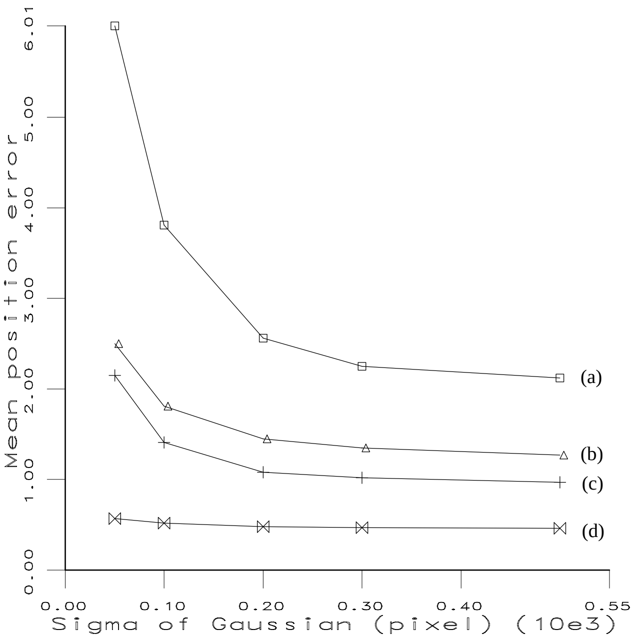

Figure 2. The curves show the mean position error versus the sigma of GBFs under four different conditions. (a) Small GBF number, low image resolution. (b) Large GBF number, low image resolution. (c) Large GBF number, high image resolution. (d) Experiment (c) and avoiding approximation errors at working space border. Generally, the error decreases by increasing the sigma, and the larger the GBF number or the higher the image resolution the smaller the position error.

Based on these experiments, we configure the GBF network such that a desired accuracy for $3D$ positions can be reached (e.g., $\pm 0.5mm$). During the application phase, first, the target object must be detected in the stereo images and the center of gravity computed from that. Second, the two $2D$ coordinate vectors are put into the GBF network for computing a $3D$ position. Finally, the robot hand will move to that $3D$ position.

4. Learning operators for recognition

For approaching and/or grasping an object its $2D$ depiction has to be detected in the stereo images. An object can be recognized in a certain image area by applying a specific function to the signal structure of that area. The output of the *recognition function* can be defined as a real value between 0 and 1, which encodes the confidence, that a certain object is depicted in the image area. Unfortunately, by changing the viewing angle of the cameras the appearance of an object changes. Regardless of variable graylevel structure of the $2D$ pattern of a target object, the recognition function should invariant compute values near to 1. On the other hand, the recognition function should compute values near to 0 for image areas depicting any other object or situation.

Regularization neural networks are used for learning and representing the recognition function. The input node represents the input pattern of the recognition function. The hidden nodes are defined by M support functions, and all these will be applied to the input pattern. This hidden layer approximates the appearance manifold of the target object, and hence the whole network can be used as recognition function. The output node computes the recognition value by a weighted combination of results coming from the support functions. The input space of the regularization network is the set of all possible patterns of the pre-defined size, but each hidden node responds significant only for a certain subset of these patterns. Unlike simple applications of regularization networks, in this application of object recognition the dimension of the input space is extremely high (i.e., equal to the pattern size of the target object, e.g., $15 \times 15 = 225$ pixel).

The approach for learning a recognition function is presented in Table 1.

Table 1. Learning a recognition function.

1. We take sample images containing the object, which has to be recognized at a later date. The samples differ from each other by a systematic change of the view conditions.
2. Optionally, we apply specific filters to the image, in order to enhance or express certain properties (see section 7).
3. From each of the (filtered) sample images, we extract a small rectangular area having the relevant object inside. The generated set of training patterns is the basis for learning the recognition function (i.e., the GBF network).
4. According to the approach for learning a GBF network, we first have to cluster the training patterns with regard to similarity.
5. Finally, we determine appropriate combination factors of the GBFs by least squares fitting using the *pseudo inverse technique*.

Steps (1), (2), and (3) will be illustrated in the sections below. The approaches of steps (4) and (5) have been described in section 2.

The learned operator for object recognition is defined by a GBF network. The collection of GBFs is based on a set of typical patterns (appearance patterns). The support of the GBFs specifies the generalizing ability for applying the operator to new patterns of the object (not included in the training set). The question of interest is: *How many GBFs are needed and which size of the support is appropriate for robust object recognition?* The robustness will be defined by incorporating an *invariance* criterion and a *reliability* criterion. The invariance criterion strives for an operator, which responds nearly equal for any appearance pattern of the target object. The reliability criterion aims at an operator, which clearly discriminates between the target object and any other object or situation. Regions of the appearance space, which represent views of objects other than the target object or any background area, should be given low confidence values.

We will experimental demonstrate a conflict in trying to maximize both criterions simultaneously. Hence, related to the overfitting/overgeneralizing dilemma (discussed above) a compromise is needed. By changing the number and the support size of the GBFs, we show the invariance and reliability performance of recognition functions.

5. Object recognition under arbitrary view angle

For learning an appropriate operator, we must take sample images of the target object under several view angles. We turn the object by using a motorized turntable and acquire orientation-dependent appearance patterns (size of the object patterns $15 \times 15 = 225$ pixel). Figure 3 shows a subset of eight patterns from an overall collection of 32. The collection is devided into a training and a testing set comprising 16 patterns each. The training set has been taken by equidistant turning angles of 22.5 degrees, and the testing set differs by an offset of 10 degrees. Therefore, both in the training and testing set, the orientation of the object varies in discrete steps over the range of 360 degrees.

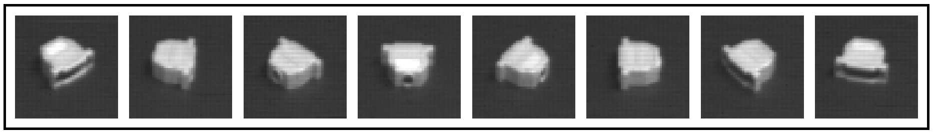

Figure 3. The target object is shown under eight equidistant turning angles. The patterns are used to learn an operator for object recognition under arbitrary view angle.

The collection of GBFs and their combination factors will be learned according to the approach of section 2. By modifying the number and/or the support of the GBFs, we obtain specific GBF network operators.

In the first experiment, a small support has been chosen, which implies a spare overlap of the GBFs. By choosing $2, 4, 8,$ and 16 GBFs respectively, four variants of GBF networks will be defined to recognize the target object. Figure 4 shows the four accompanying curves (a), (b), (c), (d) of confidence values, which are computed by applying the GBF networks to the target object of the test images. The more support functions are used, the higher the confidence values for recognizing the target. The confidence values vary significantly when rotating the object, and hence the operators are hardly invariant.

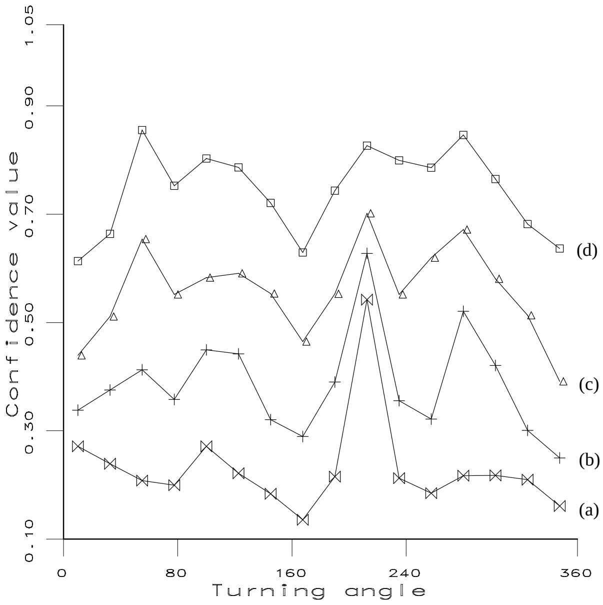

Figure 4. Different GBF networks are tested for object recognition under arbitrary view angle. The network output is a confidence value, that a certain image patch contains the object. The curves (a), (b), (c), (d) show the results under changing view angle using networks of $2, 4, 8, 16$ GBFs respectively. The more GBFs, the higher the confidence value. Due to a *small GBF sigma* the operators are *not invariant* under changing views.

The second experiment differs from the first in that a large support of the GBFs has been used, which implies a broad overlap. Figure 5 shows four curves of confidence values, which are produced by the new operators. The invariance criterion improves and the confidence nearly takes the desired value 1. Taking only the *invariance aspect* into account, the operator characterized by many GBFs and large support is the best (curve (d)).

The third experiment incorporates the *reliability criterion* into object recognition. An operator is reliable, if the recognition value computed for the target object is significant higher than those of other objects. In the experiment, we apply the operators to the target object and to three test objects (outlined in Figure 6 by white rectangles). Based on 16 GBFs as support functions, we systematic increase the support value in 6 steps. Figure 7 shows four curves related to the target object and the three test objects. If we enlarge the support of the GBFs and apply the operators to the target object, then a slight increase of the confidence values occurs (curve (a)). If we enlarge the support in the same way and apply the operators to the test objects, then the confidence values increase dramatically (curves (b), (c), (d)). Consequently, the curves for the test objects approach the curve for the target object. Increasing the support of the GBFs makes the operator more and more unreliable. However, according to the first experiments an increasing support makes the

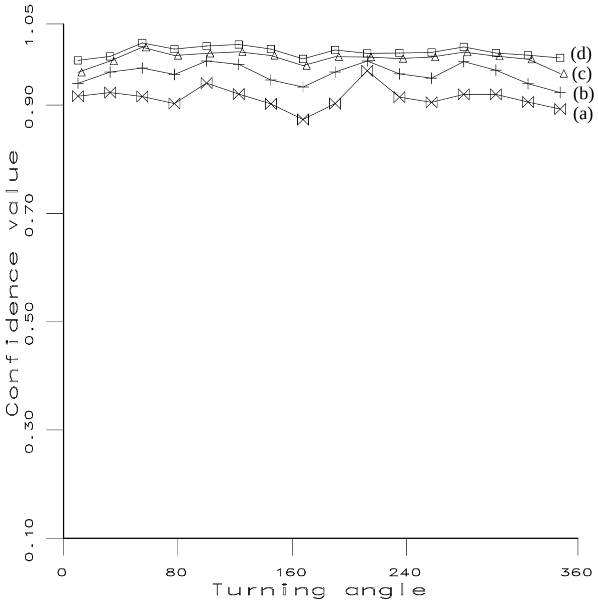

Figure 5. Similar experiments like the one in Figure 4. However, a *large sigma value* of the GBFs has been used. The learned operators respond *nearly invariant* under varying view angles.

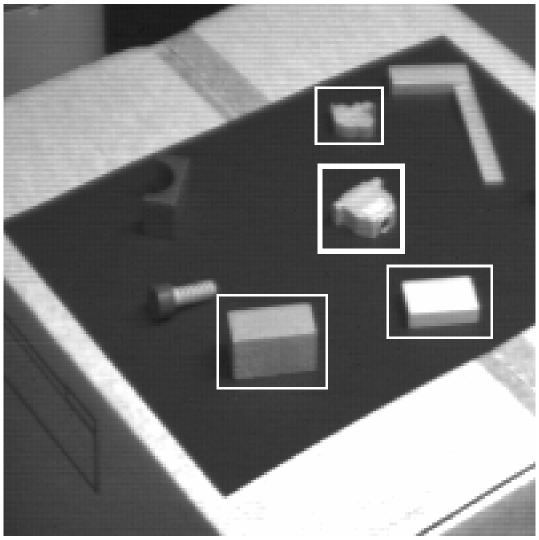

Figure 6. The image shows a certain view of the target object (in a bold rectangle) and three test objects (in fine rectangles). The GBF network for object recognition should detect the target object in this set of four candidates.

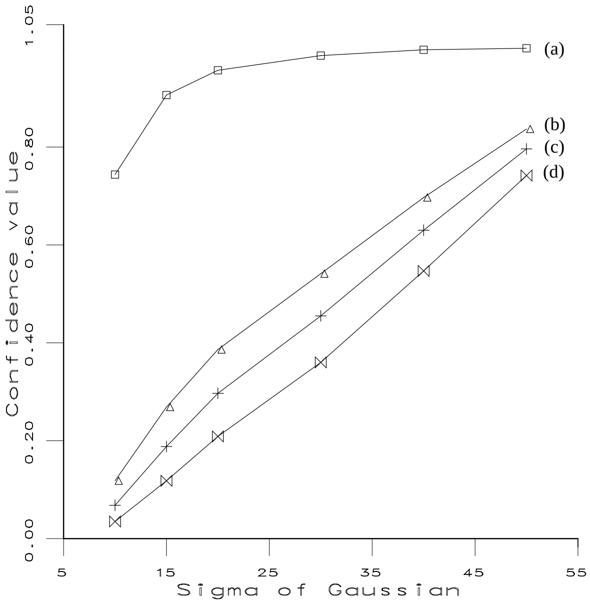

Figure 7. Six GBF networks have been constructed each with equal GBF number, but with different sigma values. Each GBF network has been applied to the image patch of the target object and to the patches of the three test objects. The GBF network computes a confidence value, that the patch contains the target object. The curves show the confidence values versus the sigma values of the GBFs. The target object (curve (a)) can be discriminated from the test objects (curves (b),(c),(d)) quite good by GBF networks of small sigma values. However, for larger sigma values the discriminating power decreases.

operator more and more invariant with regard to object orientation. Hence, a compromise has to be made in specifying an operator for object recognition.

6. Object recognition for arbitrary view distance

For learning an appropriate operator, we must take sample images of the target object under several spatial distances between object and camera. Figure 8 shows on the left the image of a scene with the target object and other objects, taken under a typical object–camera distance. On the right, a collection of 11 training patterns depicts the target object, which has been taken under a systematic decrease of the camera focal length in 11 steps. The effect is similar to decreasing the object–camera distance. The size of the object pattern changes from 15×15 pixel to 65×65 pixel. Since each training pattern encodes essential information, we define for each a single GBF (avoiding clustering). The combination factors of the GBFs are determined as before.

A further collection of 10 test images has been acquired, which differs from the training set by using intermediate values of the camera focal length. We constructed three operators for object recognition by taking small, middle, and large support of the GBFs (Figure 9).

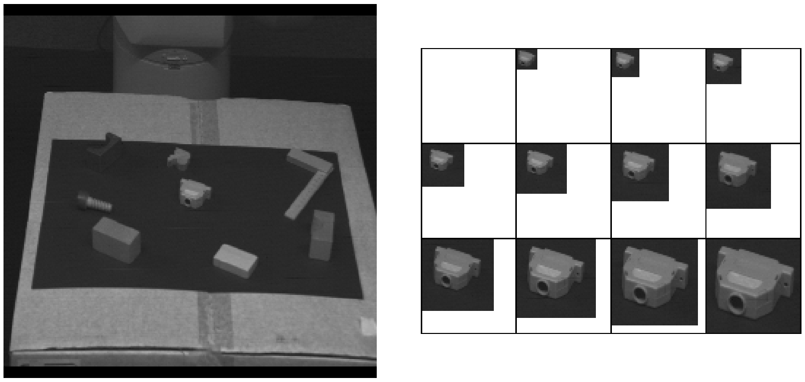

Figure 8. On the left, an image of a whole scene has been taken including the target object. On the right, a collection of 11 images is taken just from the target object under systematic increase of the inverse focal length. The effect is similar to decreasing the object–camera distance. This collection of images is used to learn an operator for object recognition under arbitrary view distance.

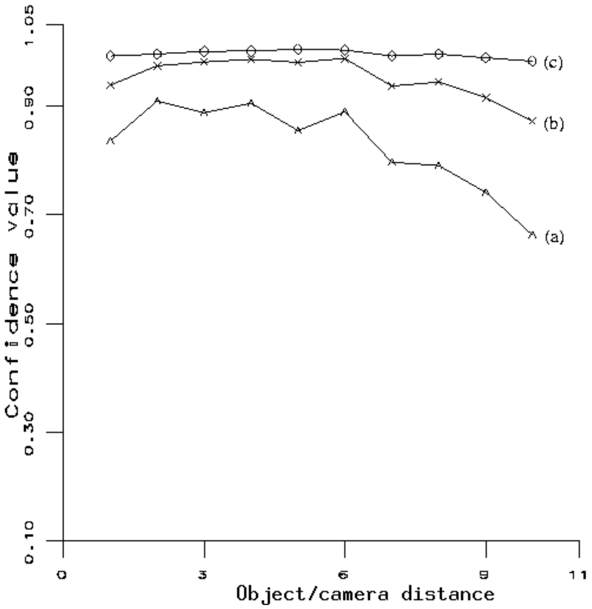

Figure 9. Three GBF networks are tested each with equal GBF number, but differing by small, middle, and large sigma value. Each network is applied to the target object in 10 test images, which differ from each other in the size of the depicted object, i.e., in the view distance. The network output gives a confidence value, that the image patch contains the target object. For small or middle sigma values (curves (a), (b)) the learned operators are hardly invariant under changing view distance. For a large sigma value (curve (c)) an invariance is reached.

In the first experiment, these operators have been applied to the target object of the test images. In curve (a) the confidence values are shown for recognizing the target object by taking a small support into account. The confidence value differs significantly when changing the object–camera distance and is far away from the desired value 1. Alternatively, if we use a middle support value, then the confidence values approach to 1 and the smoothness of the curve is improved (curve (b)). Finally, the use of a large support value will lead to invariant recognition values near to 1 (curve (c)).

In the second experiment, we investigate the reliability criterion for the three operators from above. The operators will be applied to all objects of the test image (image on the left in Figure 8), and the highest confidence value of recognition has to be selected. Of course, it is expected to obtain the highest recognition value from the target object. For comparison, Figure 10 once again depicts (equal to Figure 9) the confidence values of applying the three operators to the target object (curves (a), (b), (c)).

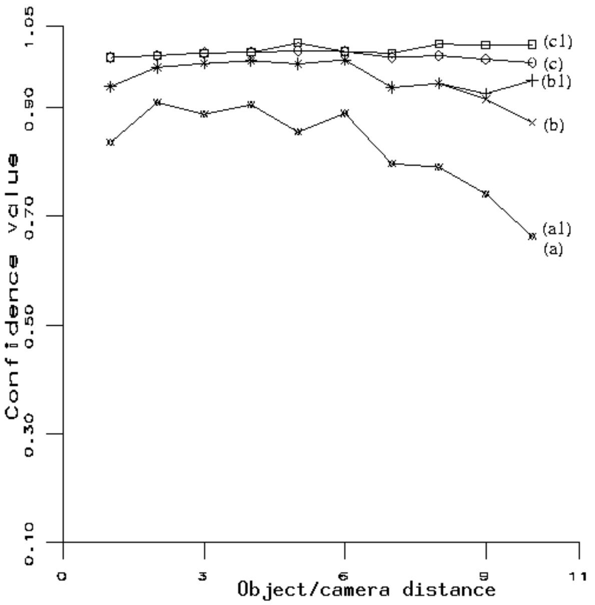

Figure 10. The curves (a), (b), (c) of Figure 9 are shown, which are the output of three GBF networks (differing by the sigma value), when applied just to the patch of the target object under varying view distance. In order to consider the reliability of these values for discriminating target and other objects the three GBF networks has been applied further to the patches of other objects under varying view distance. The left image of Figure 8 shows all these objects under a certain view distance. Each GBF network computes for each object patch an output value and the maximum of these values is taken. Repeating this procedure for all three GBF networks and for all view distances yield the curves (a1), (b1), (c1). For a small sigma value, the curves (a) and (a1) are equal, for a middle sigma value the curve (b1) surpasses curve (b) sometimes. For a large sigma value the curve (c1) surpasses curve (c) quite often. Generally, the higher the sigma value the less reliable the GBF network for object recognition.

If we apply the operator with large support value to all objects of the test images, then frequently we obtain higher confidence values for objects other than the target object (see curve (c1)). In those cases, the operator fails to localize the target object. Alternatively, the operator with middle support values meets the reliability criterion better (curve (b1) rarely surpasses curve (b)). Finally, the operator with small support values localizes the target object in all test images. The highest confidence values are computed just for the target object (curve (a) and curve (a1) are identical). Notice again the invariance/reliability conflict.

7. Recognition of grasping situations

So far, we have demonstrated the use of GBF networks for object recognition. The approach is well suited for the recognition of situations, which describe spatial relations between objects. We will exemplary illustrate specific operators for recognizing grasping situations. A grasping situation is defined to be most stable, if the target object is located between the fingers entirely. Figure 11 shows three images, each depicting a target object, two bended grasping 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 respective in front of the target object. The grasping situation in the middle image is most stable. For learning to recognize grasping stability, we moved the robot fingers step by step to the most stable situation and step by step moved off afterwards. The movement is photographed in 25 discrete steps. Every second image will be used for training and the images in between for testing.

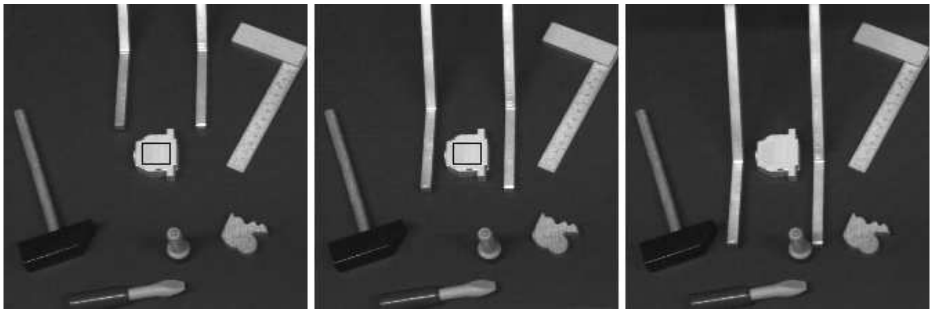

Figure 11. Three typical images of grasping situations are shown. The left and the right grasping situations are unstable, the grasping situation in the middle is stable. Altogether, a sequence of 13 training images is used, which depict first the approaching of the gripper to the most stable grasping situation and then the departure from it. This image sequence is used to learn GBF networks for evaluating the stability of grasping situations.

For learning operators, it would be possible to acquire large appearance patterns containing not only the target object, but also certain parts of the grasping fingers. However, the efficiency of recognition decreases if large-sized patterns are used. A filter is needed for collecting signal structure from a large environment into a small image patch. For this purpose, the approach in (Pauli, Benkwitz, & Sommer, 1995) used a product combination of two orthogonal directed *Gabor wavelet functions* (see fundamentals in (Rioul & Vetterli, 1991)). Figure 12 shows the overlay of two response patterns, by applying such a filter to the left and the middle image in Figure 11 and selecting the response of the (black) outlined rectangular area. A specific relation between grasping fingers and target object results in a specific filter response.

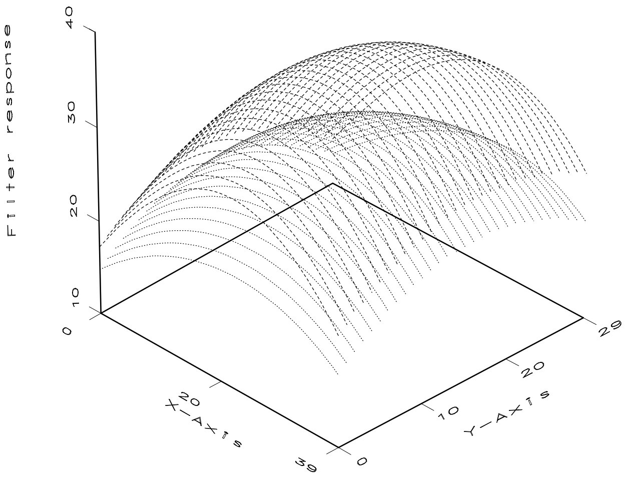

Figure 12. The product combination of two orthogonal directed Gabor wavelet functions can be applied to the image patch of grasping situations. This filter responds specific to certain relations between target object and grasping fingers. The overlay of the filter response patterns for two different grasping situations are shown. According to that, we can represent the finger–object relation by filter responses and avoid the difficult extraction of symbolic features.

Based on filter response patterns, a GBF network can be learned for situation recognition. The desired operator should compute a smooth *parabolic curve* of stability values for the course of 25 grasping situations. For the experiment, we specified many operators by taking different numbers and/or support sizes of GBFs into account. Figure 13 shows the course of stability values for two operators. The best approximation can be reached using a large number and large support of GBFs (see curve (b)).

8. Summary, discussion, and future work

Our approach of vision based robotics uses GBF networks both for camera–robot coordination and for object or situation recognition. In various experiments, it was demonstrated, how specific network configurations influence the quality of the function approximation. Depending on pre-specified thresholds for the quality, the GBF networks can be trained appropriately and then be used for online operation.

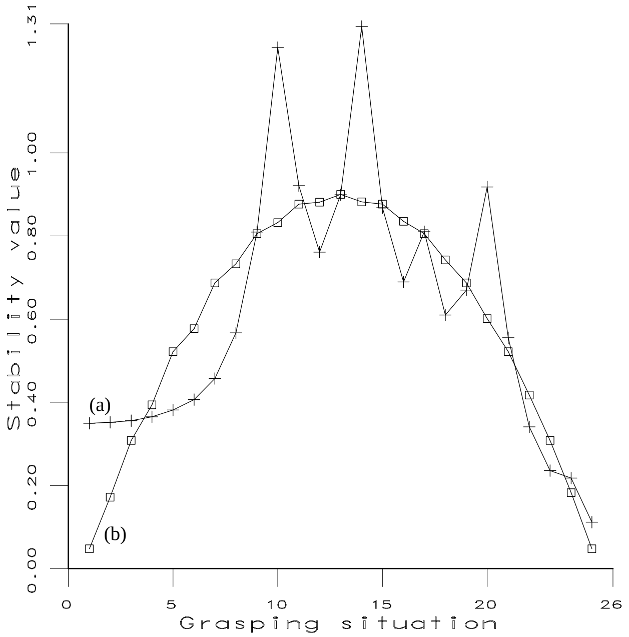

Figure 13. Based on a sequence of 13 training images (which contain the approaching to and the departure from the target object), two GBF networks have been learned. They differ mainly by a small respective high number of GBFs, i.e., from the 13 grasping situations a small respective high number of clusters are constructed. This image sequence is used for learning a parabolic curve of grasping stability where the maximum should be reached for the middle image of the sequence. Then each GBF network is applied to a succession of 25 different grasping situations depicting once again the approaching and departure. The images include both the 13 training situations and 12 test situations. If using a network with small GBF number, then the resulting course (a) of grasping stability is not the desired one. However, the course (b) resulting from the network with high GBF number is a good approximation of the desired parabolic curve. It can be used for appropriate evaluating grasping situations.

The procedure for camera–robot coordination combines three tasks, which are treated usually separately. Those are determining the camera parameters, determining the geometric relation between camera and robot, and based on that, reconstructing the position of an object from the stereo image coordinates into robot coordinates. Unfortunately, in this three-step procedure inaccuracies in parameter estimation propagate through the steps, which lead to errors in positioning the robot gripper. As opposed to that, our approach computes in one step just the relevant control vector for steering the robot gripper. In this sense, the work is similar to (Martinetz & Schulten, 1993), however with the distinction, that they define as control vector the combination of joint angles. We prefer the cartesian position and orientation of the gripper and rely on the inverse robot kinematics, which is solved quite good in our industrial robot.

Usually, the relation between $3D$ space coordinates and $2D$ image coordinates is written linear as projective transformation. This is true for an ideal pinhole camera and approximate true for a real camera lens with large focal length, but not acceptable for small focal length. For the latter, $3D$ straight lines do not project into $2D$ straight lines, because the transformation is nonlinear and leads to arcs. However, camera objectives with small focal length are able to depict a large working space, which is favourable in our applications. In our approach of using GBF networks, we can take care for such nonlinearities by spreading more gaussian basis functions into critical areas of the input space.

We have presented an approach for object recognition, which does not require a priori knowledge of the three–dimensional geometric shapes. Alternatively, the knowledge about objects is grounded in photometric appearance. As a consequence, the operator for object recognition must be learned on the basis of *raw gray levels or elementary filter responses*. Again a regularization network is used for representing and learning the operator.

It is implemented with gaussian basis functions but any other bell-shaped parabolic function is possible as well. The strength of applying the learning process to raw image data or filter responses is, that the GBF networks can generalize from a large amount of data. However, if data compression would be done prior to learning (e.g., computing symbolic values based on image segmentation), then quantization or generalization errors are unavoidable. Similary, the approach of (Ballard & Wixson, 1993) for object recognition also avoids image segmentation. A collection of *steerable filters* is applied, each responding specific to certain orientations of the gray level edges. By taking care for view variations and distances, they represent for each object a set of filter response vectors, which serve as a model base for object recognition. However, a clustering of similar model vectors to reduce the number is not treated in their work (as opposed to our GBF network approach).

Our robot vision system for object recognition has to be adjusted to the actual environment in order to reach autonomy. By doing experiments prior to the application stage (like the one presented in this work), we can later make use of the learned recognition functions. During the training stage, the regularization factor (see equation (1)) is controlled by the number and the support size of the basis functions. Various configurations reflect the well-known *invariance/reliability conflict* in object recognition. Increasing the support and/or increasing the number of GBFs makes the operator for object recognition invariant but unreliable. In order to reach a *certain degree of discriminability* between the target object and other objects, the claim for strict invariance has to be reduced into *approximate invariance*. Therefore, a further goal of doing experiments prior to the application stage is to discover an appropriate compromise between invariance and reliability of object recognition.

The greatest strength of our approach to object or situation recognition is the ability to learn (approximate) invariants under *real world changes*. Usual methods for invariant pattern recognition (Wood, 1996) have the constraint, that the permitted transformations are acting on the patterns directly. As opposed to that, in the recognition of three-dimensional objects one has to deal with changing view directions, view distances, object background, illuminations and maybe further imponderable changes. Hence, the pattern transformations are much more complicated, because they originate from real world changes. Fortunately, our experiments proved, that approximate invariants can be learned with regularization networks.

Further work is done on learning trajectories for the robot hand, so that the gripper can approach a target object along a desired route (Päschke & Pauli, 1997). For the purpose of demonstration, the operator uses the control panel to move the gripper in discrete steps to intermediate positions of the desired trajectory. A stereo camera system records this sequence, and a neural network based vision system reconstructs a smooth $3D$ trajectory. This trajectory serves as an example for a generic class of trajectories having variable starting position and orientation.

All work conducted is in line with the principle of first demonstrating the relevant objects, situations, and/or actions, and then learning from those. For the experiments in this work, the user demonstrated the coordination between the camera and the robot, the relevant target object under several viewing conditions, and several grasping situations. The system learned the camera–robot coordination, and learned to recognize the target object and to move appropriately to reach the most stable grasping position.

However, a drawback of this approach is that a large number of training examples have to be acquired to learn appropriate things. Active learning algorithms can address this problem by taking only those object views or visuomotor associations into account, which haven't been experienced so far. In the work of (Salganicoff, Ungar, & Bajcsy, 1996) an interval estimation technique is combined with classification tree construction for this purpose. A further strategy to reduce the costs of gathering training examples is to exploit general principles of image formation. A priori known invariants should be used directly instead of learning them with training samples. For example, under perspective projection the intersection of lines is invariant and the parallelism is approximate invariant. Based on a robust technique for line extraction (e.g., Hough transformation (Leavers, 1993)), we can make use of these invariants to localize polyhedral objects.

In future work, more advanced recognition functions are considered in the framework of GBF networks. It is our intention to automatically learn the variances of an object but take the known invariances into account. For example, image lines can be used for locating and describing the silhouette of an object, and then the learning approach for object recognition can be applied only within the silhouette. This strategy is a good basis for dealing with object occlusions or objects with cluttered background. Currently, our approach will be embedded in an image-based robot servoing architecture to automatically execute advanced robot tasks in complicated scenes.

9. Facilities

Sun Enterprise E4000 (4 UltraSparc processors), TRC–Bisight active binocular head, Stäubli–Unimation RX–90 robot arm (see Figure 14).

The offline training phase needs about 60 minutes for: putting the camera system in appropriate relation to the robot, taking relevant pictures from objects and grasping situations, learning the camera–robot coordination, and learning the operators for object and situation recognition. In online operation about 1 second is needed for a cycle of image taking, object/situation recognition and moving the robot hand a small increment in the direction of the most stable grasping pose.

Acknowledgement: I am grateful to G. Sommer for very useful discussions.

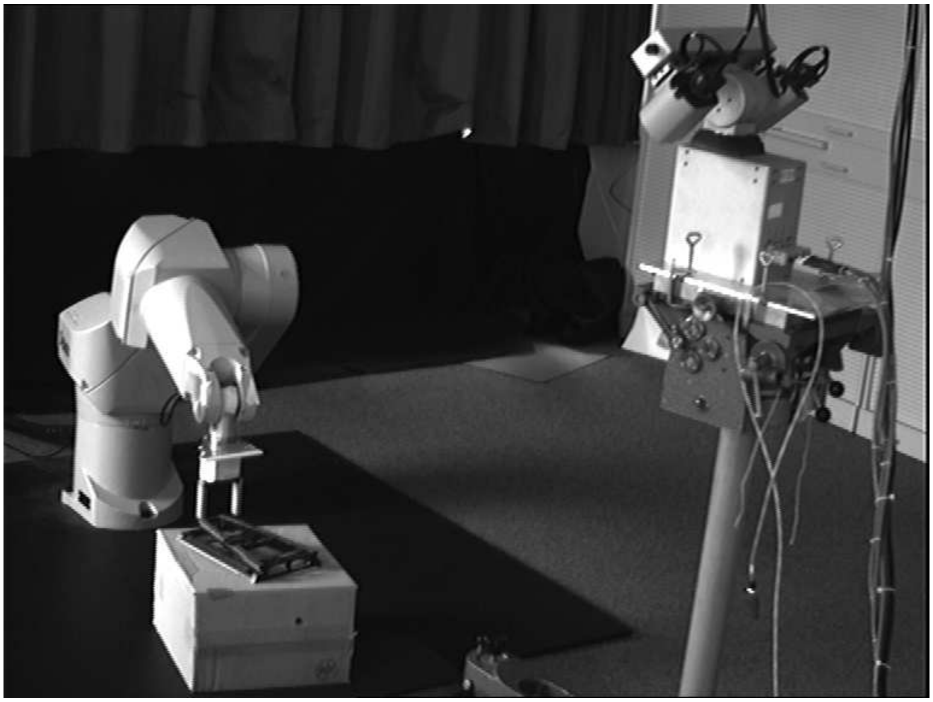

Figure 14. The image shows the robot arm and the binocular head, which have been used for the experiments of this work.

Notes

1. Tactile sensing is inevitable for a fine-tune of the robot hand, but will not be considered in this paper (Shimoga, 1996).
2. Alternative features could be detected as well, e.g., the end-effector tip.

References

Aloimonos, Y. (1993). Active vision revisited. In Y. Aloimonos (Ed.), *Active perception*. New Jersey: Lawrence Erlbaum Associates Publishers.

Ballard, D., & Wixson, L. (1993). Object recognition using steerable filters at multiple scales. *Workshop on Qualitative Vision* (pp. 2–10). New York: IEEE Computer Society Press.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. London, England: Clarendon Press.

Bruske, J., & Sommer, G. (1995). Dynamic cell structure learns perfectly topology preserving map. *Neural Computation, 7,* 845–865.

Cutkosky, M. (1989). On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation, 9,* 269–279.

Faugeras, O. (1993). *Three-dimensional computer vision*. Cambridge, Massachusetts: The MIT Press.

Kamon, I., Flash, T., & Edelman, S. (1994). *Learning to grasp using visual information* (Technical Report). Rehovot, Israel: The Weizman Institute of Science.

Leavers, V. (1993). Survey – Which Hough transform ? *Computer Vision and Image Understanding, 58*, 250–264.

Martinetz, Th., & Schulten, K. (1993). A neural network with Hebbian-like adaptation rules learning visuomotor coordination of a PUMA robot. *International Conference on Neural Networks (ICNN)* (pp. 820–825).

Maxwell, B., & Shafer, S. (1994). A framework for segmentation using physical models of image formation. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 361–368). Seattle, Washington: IEEE Computer Society Press.

Murase, H., & Nayar, S. (1995). Visual learning and recognition of 3D objects from appearance. *International Journal of Computer Vision, 14*, 5–24.

Päschke, M., & Pauli, J. (1997). Vision based learning of gripper trajectories for a robot arm. *International Symposium on Automotive Technology and Automation (ISATA)* (pp. 235-242). Florence, Italy: Automotive Automation Limited.

Pauli, J., Benkwitz, M., & Sommer G. (1995). RBF networks for object recognition. In B. Krieg-Brueckner & C. Herwig (Eds.), *Workshop Kognitive Robotik*. (Technical Report). Bremen, Germany: Universität, Zentrum für Kognitive Systeme.

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

Press, W., Teukolsky, S., & Vetterling, W. (1992). Numerical recipes in C. Cambridge, Massachusetts: Cambridge University Press.

Rioul, O., & Vetterli, M. (1991). Wavelets and signal processing. IEEE Signal Processing Magazine, October, 14–38.

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

Salganicoff, M., Ungar, L., & Bajcsy, R. (1996). Active learning for vision-based robot grasping. *Machine Learning, 23*, 251–278.

Schalkoff, R. (1992). *Pattern recognition - statistical, structural, and neural approaches*. New York: John Wiley and Sons.

Shimoga, K. (1996). Robot grasp synthesis algorithms - A survey. *The International Journal of Robotics Research, 15*, 230–266.

Trobina, M., Leonardis, A., & Ade, F. (1994). Grasping arbitrarily shaped objects. *Mustererkennung 1994* (pp. 126–134). Wien, Österreich: PRODUserv.

Utgoff, P. (1986). *Machine learning of inductive bias*. Hingham, Massachusetts: Kluwer Academic Publishers.

Wood, J. (1996). Invariant pattern recognition - a review. Pattern Recognition, 29, 1–17.

Contributing Authors

J. Pauli studied Computer Science at the Technische Universität München and attained his masters degree in 1986. From 1987 to 1992 he was a staff member of that Computer Science institute. During this period, his interest was on symbolic machine learning and computer vision. He finished his doctors degree with a thesis on *Explanation-based computer vision for image sequences* (in german). Since 1993 he is a staff member of the Computing Science institute at Christian-Albrechts-Universität zu Kiel. His research area is on autonomous robots, computer vision, and neural network learning.