Proc. Int. Joint Conf. on Neural Networks 2000, (5):137-142

Learning Fine Positioning of a Robot Manipulator based on Gabor Wavelets

Jörg Walter · Bert Arnrich · Christian Scheering

Department of Computer Science · University of Bielefeld
D-33501 Bielefeld · Email: walter@techfak.uni-bielefeld.de

Abstract: A system for learning the pre-grasp positioning task for a robot manipulator is presented. The images delivered from a gripper mounted camera are analysed using Gabor filters which resemble the spatial response profiles of receptive fields found in visual cortex neurons. Using a quite small feature set, the system demonstrated efficiency with respect to speed and accuracy, as well as robustness against changing light conditions.
Furthermore, we compare it to two other approaches, aiming at the same goal: an appearance-based PCA fuzzy control and a PSOM based Hough-Transform system.

1 Introduction

Neural networks were evolved by nature to enable perception and action. Therefore, artificial neural networks seem as an appropriate choice for learning one of the most demanding sensor-based manipulation skills – grasping. One of the main obstacles for industrial applications is the availability of robust and inexpensive sensor–action systems. As the price and size of reasonable vision systems is decreasing, camera sensors become a popular expansion to robot system. Traditional robot vision research focuses on the use of explicit world models and how to build them from raw sensor data. Scene reconstruction is undoubtedly useful, but is expensive and often to complex in a changing real-world environment. We think that locally operating learning schemes are the best candidates to advance intelligent robot systems. What are the best feature extraction plans to feed a learning network?

Gabor wavelets proved advantageous for object and face recognition [4], acknowledged for their pose invariance. Here, we examine the reverse task. Knowing the object, how well can we estimate the pose? What are the best features to built a working sub-system? How do they compare to other approaches?

System Overview: The overall aim of our system is the structure assembling demonstrated using a set of wooden pieces including nuts, screws, ledges, and cubes, see Fig. 1. This task can be divided into several smaller ones: a single part has to be identified, the gripper is brought in a suitable pre-grasp position, the target is firmly enclosed and gets finally transfered to the desired mating/assembly position with other parts. The robot system consists of a 6 DOF manipulator (Puma 260) with a camera attached to the parallel yaw gripper.

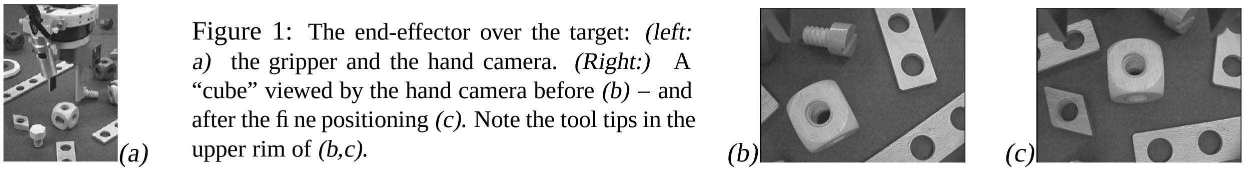


Figure 1: The end-effector over the target: *(left: a)* the gripper and the hand camera. *(Right:)* A "cube" viewed by the hand camera before *(b)* – and after the fine positioning *(c)*. Note the tool tips in the upper rim of *(b,c)*.

Grasping without any alignment help requires that the objects are picked with certain precision. Failures include the risk of *(i)* object–gripper collisions, *(ii)* pushing/displacing something before yaw closure, and *(iii)* bad object-in-gripper alignment (creating trouble for later part mating).

Here we discuss the critical pre-grasp phase, i.e. the 3 DOF fine-positioning of the manipulator after an initial coarse positioning has been completed. This implies that the resting object is visible inside the viewing angle of the hand camera and its type and vertical position is known. Now the system has to deal with significantly changing appearance of the target objects with respect to *(i)* the local lighting situation (occlusion of lamps by the robot itself, interfering humans, etc.), *(ii)* image contrast and color (several possible objects), *(iii)* parallax effects by the camera viewing from a close and tilted position.

2 Object Representation

In order to efficiently employ a learning neural network for pose estimation we need a suitable object representation gained from the sensory input, in our case a camera image. “Suitable” means here, the feature set is of minimal size providing the desired accuracy and in the following faster learning with fewer neurons.

2.1 Object Representation With Gabor Filters

Biology gave us inspiration: 1987, Jones and Palmer [3] showed by cat visual cortex experiments, that receptive fields of simple cells fit well to a profile model previously suggested by Daugman 1980 [1]. This model describes the spatial sensitivity by a 2D extention of Gabor’s work (1946, originally in the time domain). By a local formulation of the frequency content he created a “localized” Fourier analysis, here written as a complex kernel function:

$$\Psi(x,y) = \Psi_{\alpha\sigma\lambda}(x,y) = \exp\left(-\frac{x^2+\alpha^2y^2}{2\sigma^2}\right)\left[\exp\left(-2\pi i\frac{x}{\lambda}\right) - \exp\left(-\frac{\sigma^2}{2\alpha}\right)\right] \tag{1}$$

describes a Gaussian bell function – modulating a planar wave. The wave has the period length λ in x-direction; the elliptical Gaussian has a longitudinal width σ and σ/α transversal (aspect ratio α). The last, constant term make the filter DC-free, hence invariant to any shift in gray-level of the image.

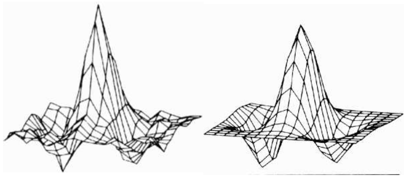


Figure 2: The 2D Gabor filter *(right)* fits simple cell spatial response profile *(left)* of receptive fields in cat striate cortex neurons [3]. See also Fig. 6.

Equation 1 can be called *mother wavelet* and a complete family of self-similar *daughter* wavelets (sometimes called *jet*) can be constructed by the generating function

$$\begin{aligned} \Psi_{mpq\theta}(x,y) &= 2^{-2m}\Psi(x',y') \\ x' &= 2^{-m}[+x\cos\theta + y\sin\theta] - p \\ y' &= 2^{-m}[-x\sin\theta + y\cos\theta] - q. \end{aligned} \tag{2}$$

Here the substituted variables incorporate dilations of the wavelet in size 2^{-m}, translations in position (p,q), and rotations through the angle θ.

Each cell’s receptive field can be modeled by a Gabor wavelet function, parameterized by the center (p,q), the wavelength λ ($m=0$) in direction θ with Gaussian elliptic envelope (with width σ and σ/α) and a complex phase angle ψ (projecting a mixture of the real and imaginary part).

Our system uses a collection of n those artificial neurons, all looking at the same image but each with a different receptive field. Thus the seen object gets represented by n values, i.e. the scalar product of the image by the appropriate Gabor filter mask. For a larger group of neurons, which differ only in their center position (p,q), the procedure can be speeded up by performing a convolution and implementing it as a product in the 2D-Fourier space.

2.2 Object Representation Using Eigenimages and PCA

An other compact object representation, based on the object’s appearance, became popular [5, 9]. Here, the Principal Component Analysis (PCA) technique determines the *eigenimages* of a collection of c object images. Each image i, e.g. of size $n\times n$ pixel, is considered as n^2-dimensional vector $\mathbf{x}_i$. The ($n^2\times n^2$ dimensional) covariance matrix C is defined as $C = 1/c\sum_{i=1}^{c}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$ with the mean image $\bar{\mathbf{x}} = 1/c\sum_{i=1}^{c}\mathbf{x}_i$. The *eigenimages* are the eigenvectors $\mathbf{e}_j$ of the (symmetric and positive) matrix C. The $\mathbf{e}_j$ corresponding to the m largest eigenvalues span a new orthogonal base representing the principal components of the initially analyzed image collection (see Fig. 8). Even for small images, with e.g. 10000 pixels, the matrix algebra requires special algorithms or special network architecture in order to remain computationally tractable [5, 6]. A new image $\mathbf{x}$ is then represented by the m scalar products $\mathbf{p} = (\mathbf{e}_1^T\mathbf{x}, \mathbf{e}_2^T\mathbf{x}, \ldots, \mathbf{e}_m^T\mathbf{x})^T \in \mathbf{R}^m$.

2.3 Object Representation Using the Hough Transformation

The Hough Transformation (HT) is a classical image analysis method for curve detection. It uses the duality between points on a curve and parameters of that curve. In principle any analytical curve can be detected, but in reality the computational costs grow exponentially with the number of parameters required [2]. As outlined in [8], we chose to stay with the simplest curve: a straight line with two describing parameters, e.g. angle θ (to the x-axes) and its normal distance d to the origin (0,0). A discretized (d, θ)-region of interest – also called the *accumulator* – is chosen and the image is systematically searched for pixels contributing to a given line. The usual procedure is to binarize the result of an edge-detection filter, as illustrated in Fig. 3.

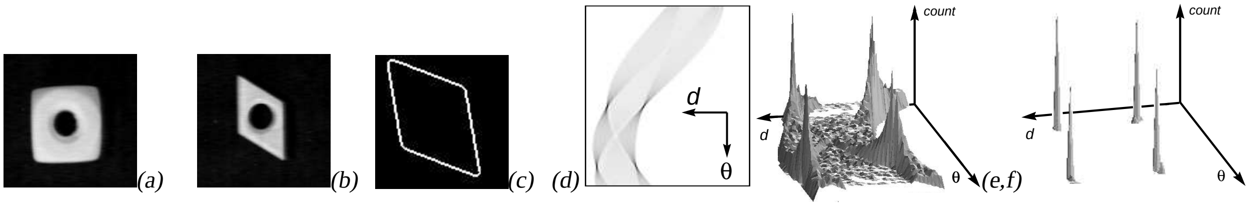


Figure 3: *(a,b)* Two objects seen by the hand camera here with vertical viewing angle. *(c)* The binarized edge-preprocessed nut image which becomes Hough transformed: each pixel corresponds to a sinusoidal curve in the Hough parameter space and increases all accumulator cells on this path. *(d)* shows the accumulator as intensity image and *(e)* as 3D rendering. HT-Curves from pixels lying on a line intersect in one HT-point, resulting in high counts. *(f)* Applying the so-called "back-transform" [2] algorithm sharpens the peaks and makes their localization more secure. Mapping to the rotation displacement is straight-forward since each HT-peak codes for one line in image space. In the optimal case, the four (HT-peak) identified lines mark exactly the outline of the target object.

3 Experimental Setup and the Gabor System in more Detail

The "Cubical" Challenge: For comparison reasons we focus in the following on one target object, the cubical wooden piece already shows in Fig. 1. Its size is little smaller than the open gripper and calls therefore for grasping tolerance of about 2 mm and 5-8°. The wooden "cube" is challenging for machine vision: due to the widely rounded corners and the three axial screw holes (creating shadow), its facial surfaces are actually ring shaped. The contour, seen from an tilted viewing angle, resembles an egg with moving bumps when rotating the "cube". One way to avoid this problem, is to turn the end-effector such, that for image grabbing the camera is looking vertically downwards (as the Hough system actually does, see Fig. 3a; compare 4). The price is a robot transfer delay (decrease of operation speed) and extra kinematic restrictions within the robot's workspace.

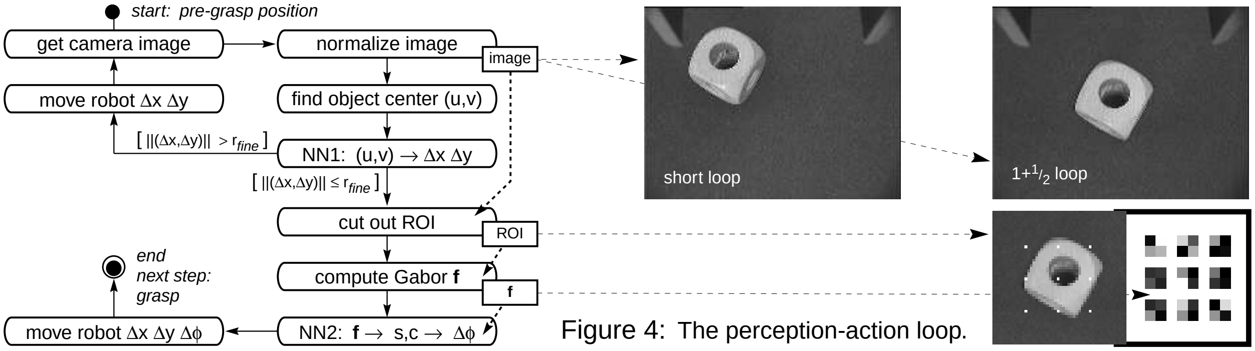


Figure 4: The perception-action loop.

Pre-Grasp Procedure: In order to allow vivid and accurate operation we keep the gripper vertical and subdivide the fine positioning in two parts: *(i)* a fast and coarse – pure translational part and *(ii)* a rotational/translational fine part. Fig. 4 displays the simplified procedure in an UML-activity diagram. The grabbed image is preprocessed and the object's center of gravity in image coordinateds (u, v) mapped by the first neural network (MLP) to the Cartesian translational command $(\Delta x, \Delta y)$, required to move the robot over the target object. If the displacement is too large – therefore the parallax effects too disturbing – the robot moves first and looks again (top loop, $r_{fine} = 4\,mm$). The rotational adjustments is determined from an (u, v)-centered region of interest (ROI with size 50×50 pixel). A small set of features $\mathbf{f}$ is extracted and a second neural network (also MLP) maps to the shortest rotational approach command $\Delta\phi$. Executing the robot move command prepares the system for the next step, which is usually the force-torque guarded grasping of the object. Alternatively, the loop can be closed for testing purposes.

Preprocessing: The grabbed color image of size 192×144 pixel is reduced to one channel by a pixel-wise maximum selection in the R,G, and B channel. Then the mean and standard deviation of all pixel values is computed and a global linear pixel intensity transformation is applied which normalizes the image to the training standard conditions (i.e. the same intensity average and variance).

Object Localization: The normalized image is binarized and a standard blob detection algorithm selects the object center. If the image is not cluttered, a fast row and columnwise histogramming is sufficient. The first neural network NN1, a 2-3-2 resilient backprop accelerated MLP, maps to the desired translational correction $(\Delta x, \Delta y)$, which is used in the Cartesian transfer command and send to the robot.

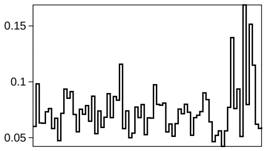
Object Rotation and Angle Wrapping: The second neural network (NN2) has to code for the shortest rotational correction $\Delta\phi$. Here occurs the problem of angle wrapping ($\phi = \phi \pm 360°$) and the rotation object symmetry count κ, e.g., for the cube $\kappa = 4$ (i.e. same appearance for $\Delta\phi = 0°$, 90°, 180°, 270°). We solve this with an angular sine/cosine pair encoding for the MLP-output layer

$$s = \sin(\kappa\Delta\phi), \quad c = \cos(\kappa\Delta\phi), \quad \text{and the back-transformation} \quad \Delta\phi = \frac{1}{\kappa}\,\text{atan}\left(\frac{s}{c}\right). \tag{3}$$

Selecting the Training Data Set: The desired nominal grasping position is *"demonstrated"* to the system by guiding the robot *once* (e.g. via a 3D-mouse) in the correct pose and defining the approach distance. Starting from there, a set of images is automatically aquired. An image gets grabbed from the hand camera after displacing the robot by the value $-(\Delta x, \Delta y, \Delta\phi) \in [-a, a] \times [-a, a] \times [-b, b]$ (for NN1 $a = 25\,mm$ and NN2 $a = 5\,mm$, $b = 180°/\kappa$; the training set is sampled from a 3×3×7-grid, while the test set is randomly sampled). The image processing results in association with the desired robot command $(\Delta x, \Delta y)$ for NN1, and $\Delta\phi$ for NN2, provide the training data for the supervised learning phase of the neural networks.

Selecting the Gabor Filter Set: Since the optimal feature set for our task was unknown, we carried out some systematic simulation tests with varying Gabor filter combinations, see Fig. 5.

Figure 5: Gabor Feature Selection: Shown is the obtained RMS positioning accuracy for various Gabor filter combinations. Each was a simulation experiment repeated 10 times including a training phase of NN2 and an evaluation phase for the entire system. The best experiment's filter set is displayed below.



The winning configuration was a surprise: it consists of only nine different center positions on a 3×3 grid centered in the $(50 \times 50\,pix)$ image ($p, q \in \{13, 25, 27\}\,pix$). At each position we centered four even Gabor filters with $\lambda = 30\,pix$, $\sigma = 12.5\,pix$, and $\theta \in \{0°, 45°, 90°, 135°\}$ as depicted in Fig. 6. The first surprise was, that already 36 image features are sufficient for the rotation estimation task. The second was, that the system does not prefer higher Gabor frequencies which would be more sensitive to the edge positions.

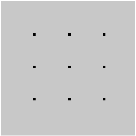
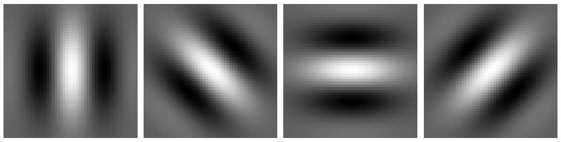


Figure 6: *(Left:)* Locations of the receptive fields centers (p, q) in the ROI-image. *(Right:)* Gabor filter set with four orientations, shown only for the middle location. The other 32 filters are shifted versions. See also Fig. 2



4 Experimental Results – a Comparison

4.1 The Gabor based System

Accuracy: For the Gabor system we achieved an asymptotic RMS positioning accuracy of $0.08\,mm$, $0.2\,mm$ (in x, y direction) and $0.8°$ (in ϕ) after a couple of iterations in the fine-positioning loop in Fig. 4. This is far more than

Table 1	**Gabor System**	**PCA System**	**Hough System**
Accuracy with *Good* Illumination	(0.5, 0.7, 6.6°) after $1\frac{1}{2}$ loops		(4 DOF: x, y, z, ϕ)
$(x/mm, y/mm, \phi/°)$	(0.1, 0.3, 1.2°) after $2\frac{1}{2}$ loops	(0.4, 0.7, 0.6°)	(0.5, 0.8, 1, 1.4°)
Save Grasp requires	$1\frac{1}{2}$ loops	5 loops	$2\frac{1}{2}$ loops
Accuracy with *Poor* Illumination	(0.2, 0.2, 1°) after $6\frac{1}{2}$ loops	(3.1, 1.0, 6.1°)	—
Save Grasp requires	$3\frac{1}{2}$ – $5\frac{1}{2}$ loops	20 loops	*[miracle]*
Controller	2 × MLP	4 × Neuro-Fuzzy	PSOM + Model
ϕ Feature Generation by	36 general filter masks	3 eigenimages	line HT
Total Time per Loop Iteration	1–2 sec	1–2 sec	1–2 sec
Training Time	$<$ 1 min	3 hours	$<$ 1 min

required. For reasonably good illumination conditions one short – pure translational (termed "half" loop) – and one full positioning loops is sufficient (half+full="$1\frac{1}{2}$"), see Tab. 1.

Robustness – Illumination: Changing local lighting conditions are a real threat to many vision based control algorithms. The described preprocessing method proved quite efficient: with stepwise dimmed lights and furthermore changing to a single sideward lamp (producing bad cast-shadows), we found that the performance degraded only in the speed of convergence. The basic operation was stable up to extremely poor illumination conditions.

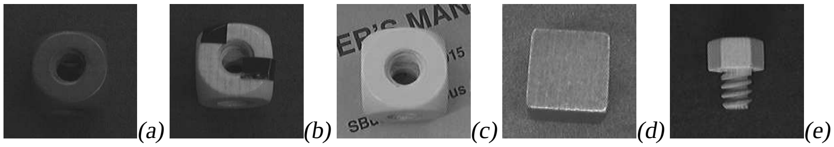


Figure 7: *(a-c)* Poor image conditions – successfully mastered. *(d-e)* other target objects (block *(d)* used the same NN2, the screw *(e)* a different).

Different Object Colors and Background: The training object (yellow cubical) shows very good contrast to the gray table surface. But the fine-positioning system works without modifications also for comparable objects, for other colors, and under bad conditions, e.g. poor brightness contrast and dim light. As Fig. 4.1 displays, partly covered objects or even textured background did not bring the grasping system to tumble.

4.2 The PCA–Fuzzy Control System [7, 9]

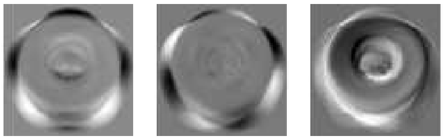


Figure 8: The first three *eigenimages* $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ serve as image feature mask for the rotational part of the PCA-system (see also Sec. 2.2) [7].

The robot commands were trained and later recalled by a B-spline neuro-fuzzy controller (FC). Since the calculation and processing of eigenimages is quite expensive, only three ($m = 3$) were computed for the rotational part. Since only 3 features are extracted for a new image, the system has to repeat the positioning loop more often in order to achieve the grasp precision. The total training time is more than three hours (a day for $m = 4$, weeks for $m = 6$). The system is robust against change in object color and illumination but speed *and* positioning accuracy degraded with decreasing illumination conditions, see Tab. 1.

4.3 The Hough Transformation System

This system additionally learned the object height (4 DOF) and employed a system setup with a larger 560 Puma robot and a three-fingered articulated hydraulic hand for grasping. We used a 3×3×3 PSOM to learn the translations from 27 images and a object model to validate the gained HT-peak information [8]. The Hough Transform method is able to extract (in comparison) the most information out of a good image. For the HT, a good image contains only clear polygonal edges. This requires a significant amount of image preprocessing including smoothing, edge-filtering, binarization, segmentation, and masking tricks to remove disturbing edges inside the object contour (see empty outline in Fig. 3c). However, the system works very well under normalized conditions – with good object-background contrast, no clutter, good illumination, and a viewing angle (strictly vertical) which lets the rounded corners disappear.

5 Summary and Conclusion

We presented a pre-grasp fine positioning scheme for a robot camera-in-hand system. It employs a small set of only 36 Gabor masks probing the image for spacial frequency content at nine locations and four orientations. The feature set is universal for a family of similar objects and can be easily adapted and/or enriched for a broader spectrum of shapes. The preprocessing stage performs an image intensity adaptation and guidance of the rotational sensor (ROI). The overall system is robust with respect to the image condition, e.g. changes in illumination and some amount of occlusions and clutter.

We compared the described system to previous work implemented in the same lab, pursuing the same goal – but applying different techniques.

The PCA and neuro-fuzzy controlled approach [9] uses a even smaller feature set based on eigenimages specialized for the appearance of one single object type (Fig. 8 shows three). The small number is mainly a compromise to balance between the exponentially growing training time [7] and the information gained. Of course, as more useful information the system can extract, as fewer iterations it needs for precise grasping. The main disadvantage of the PCA-FC system is the limited training and performance speed. On the other hand, the PCA-system display reliability and robustness.

The Hough Transform approach employs classical image processing techniques [8]. It delivers good results under highly normalized conditions. The main reason for the poor robustness is the focus on differential image information (edges), the associated noise sensitivity, and the information loss in the binarization step. The countermeasures are expensive (e.g. contour following, region growing, etc.) and do not fundamentally solve the problem. Furthermore, a model of the expected HT-peak constellation is needed, in order to monitor and guide the peak-detection algorithm.

Summarizing, the Gabor-filter based system uses very favorably image processing techniques, working also in visual cortex neurons. Employing only 36 “simple cells” we built a technical system which was *(i) calibration free* (e.g. no camera calibration required), *(ii) direct* (no expensive image processing like segmentation, region growing, etc.), and *(iii) fast* (Gabor feature detection and ANN mappings could by implemented in real-time). It demonstrated *(iv)* the *robustness* real-world capable system require.

References

[1] John Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20:847–856, 1980.

[2] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, Image Processing*, 44:87–116, 1988.

[3] Judson Jones and Larry Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58:1233–1258, 1987.

[4] M. Lades, J. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R. Würtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42:300–311, 1993.

[5] Hirosi Murase and Shree Nayar. Visual learning and recognition of 3-d objects from appearance. *Int. Journal of Computer Vision*, 14:5–24, 1995.

[6] Terry Sanger. An optimal principle for unsupervised learning. *NIPS*, 1989.

[7] Ralf Schmidt. Personnel communication; images courtesy of.

[8] Dirk Schwammkrug, Jörg Walter, and Helge Ritter. Rapid learning of robot grasping positions. In H. Araujo and J. Dias, editors, *Proc. 7th Int. Symp. on Intelligent Robotic Systems (SIRS)*, pages 149–155, July 1999.

[9] Jianwei Zhang, Ralf Schmidt, and Alois Knoll. Appearance-based visual learning in a neuro-fuzzy model for fine-positioning of manipulators. In *Proc. Int. Conf. on Robotics and Automation (ICRA-99)*, 1999.