Visual Checking of Grasping Positions of a Three-Fingered Robot Hand

Gunther Heidemann and Helge Ritter

Universität Bielefeld, AG Neuroinformatik, Germany
gheidema@techfak.uni-bielefeld.de

Abstract. We present a computer vision system for judgement on the success or failure of a grasping action carried out by a three-fingered robot hand. After an object has been grasped from a table, an image is captured by a hand camera that can see both the object and the fingertips. The difficulty in the evaluation is that not only identity and position of the objects have to be recognized but also a qualitative judgement on the stability of the grasp has to be made. This is achieved by defining sets of prototypic "grasping situations" individually for the objects.

1 Introduction

Grasping objects with robot grippers or even anthropomorphic artificial hands is one of the most challenging subjects in current robotics. Up to now, the abilities of robotic systems in grasping objects are surpassed by human skills by far: We can grasp objects of almost arbitrary shape, many different sizes and we are able to adapt easily the applied forces to light or heavy weight. This ability relies on a complex interaction of the controlling neural system and the "sensors", which are mainly haptic and visual. Especially force sensing in the fingertips plays an important role. Unfortunately, there is no technical equivalent to human fingertip force sensing by now, though progress has been made e.g. using piezo-resistive pressure sensitive foils [6] for tactile imaging [5,9] or piezo-electrical pressure sensitive foils [1] for dynamic sensing [2,14].

Though it is no problem to provide high quality sensors in general, sensors the size and shape of a human fingertip are extremely difficult to produce (and apply). In our experience such fingertip sensors are still unreliable and coarse in measurement. Consequently, other sensory sources have to be exploited. A suitable means is visual control as miniature cameras and image processing hardware are easily available. Our approach is therefore to supplement fingertip sensing by a visual check of the points where the fingers touch the object.

In this contribution we deal with the evaluation of images captured by a hand camera which can see the grasped object and the fingertips. I.e., the object has been grasped already – this can verified by sensors –, but it is still unknown how stable the object is held. Evaluating this "grasping situation" is a challenge to computer vision since it is not a classical classification or pose estimation task but instead a complex situation has to be judged. Moreover, the relative

G. Dorffner, H. Bischof, and K. Hornik (Eds.): ICANN 2001, LNCS 2130, pp. 891–898, 2001.

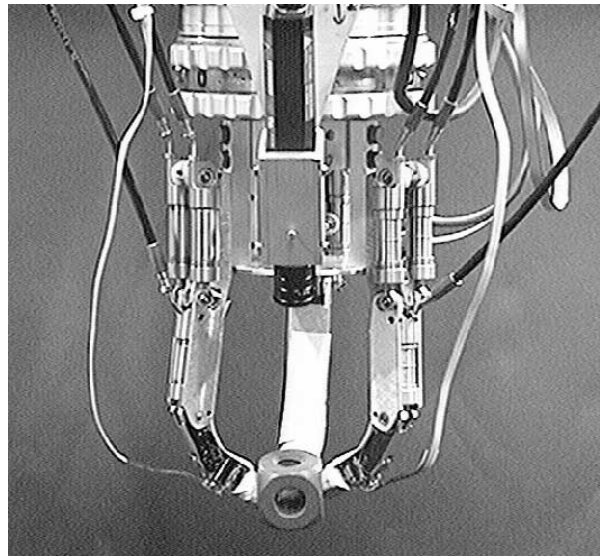

Fig. 1. The three-fingered hydraulic hand holding an object. A camera in mounted in front, looking at the object and the fingertips.

fingertip positions on the object are sometimes even difficult to see for humans (Fig. 2). In our approach, these problems are solved defining typical success- or failure-situations known from robotic experience as classes for a neural classifier, thus achieving a qualitative judgement on the situation.

2 The Robot Setup

Our system is based on a standard industrial robot arm (PUMA 500) with six degrees of freedom. The end-effector is an oil-hydraulic robot hand developed by Pfeiffer et al. [10]. It has three equal fingers mounted in an equilateral triangle, pointing in parallel directions (Fig. 1). Each finger has three degrees of freedom: bending the first joint sideways and inward at the wrist, and bending the coupled second and third joint. The oil pressure in the hydraulic system serves as a sensory feedback. A detailed description of the setup can be found in [7]. Fig. 1 shows also a force/torque wrist sensor which will not be used in these experiments.

Currently the robot system is used within a man-machine interaction scenario where a human instructor tells the robot to grasp an object on a table. An active stereo camera head apart from the robot detects the hand movements of the instructor as well as the objects. The world coordinates of the indicated object are given to the robot system as soon as a pointing gesture could be detected. The robot arm then carries out an approach movement until the hand camera

detects the object to control the final approach. The object is then grasped by the three-fingered hand. This system is described in [11].

Up to recently, the only feedback on the success of the grasping movement was the sensed hydraulic oil pressure. By this data, however, it can only be judged if the object is still in the grasp or if it was lost. Though there are additional position sensors on the hydraulic motor pistons, these position measurements cannot be used to estimate the resulting finger positions within the required accuracy due to mechanical hysteresis.

Consequently, a supplementary system has been developed which checks the grasping position visually using the hand camera. We will first characterize the task of this system by some examples. Fig. 2 shows a cube shaped object with a hole at each side in the grasp of the hand. The row above shows as an overview the hand from the side, below are the corresponding views of the hand camera. Besides checking if the object is in the hand at all, the questions relevant to robotics which the hand camera should answer are the following:

1. In which position or pose is the object?
2. Which of the fingertips have supporting contact?

It should be noted that both these questions can only in part be answered in general, like that e.g. three contact points give usually more stability than two. However, whether a certain fingertip position does really support the object or not depends on the type of the object. Similarly, the absolute position of the object held in the hand is not necessarily relevant for all tasks. Whether a position is useful depends on the further proceeding, e.g. putting the object down or attaching it to other objects. Therefore, object and task specific knowledge must be introduced by defining appropriate judgement-classes to be trained by the recognition system. This will be outlined in section 4.

3 Visual Recognition System

The recognition system for the hand camera is composed of two modules (left and right in Fig. 3): (a) a simple location module to find the fingertip regions and (b) a classification module (VPL-classifier, section 3.3) of which two instances (VPL1 and VPL2) are used: one for the classification of the grasped object and one for the evaluation of the three fingertip regions.

Image processing is carried out on two resolutions (images I_1 and I_2, not shown in Fig. 3): The camera input is a grey level image sub-sampled to 192×144 pixels (image I_1), which is used as input for the classifier. I_2 is an even further sub-sampled version of the camera image of 96×72 pixels, it is used for ROI selection.

3.1 Locating the Robot Fingertips

In order to keep ROI selection as simple and fast as possible, we use black background and the robot fingers are covered with white tape. The objects have

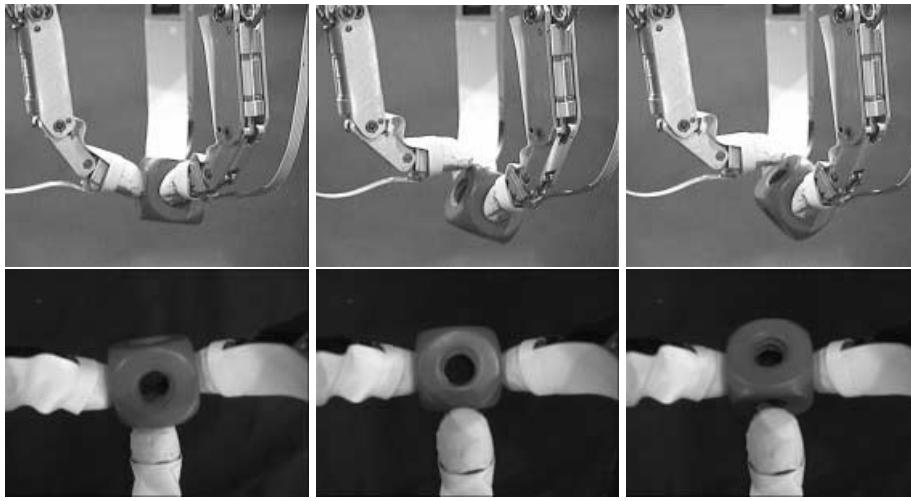

Fig. 2. Three grasping positions, above view from the side, below view of the hand camera. Left: Object held correctly by all three fingers. Middle: Object held only by two fingers (left finger has lost its grip) but still stable and in similar position like (Left). Right: Like (Middle), but object in a different position. Note how small the differences appear from the position of the hand camera.

intensity values between those of the background and the fingers, thus the finger-regions can be found by a simple intensity based binarization: Starting with a high threshold $\vartheta = \vartheta_{start}$ ($\approx$ finger intensity) on image I_2, ϑ is decreased until three blobs of a minimum size and a priori defined form features can be found by a connectivity analysis of the binarized image. Since the fingers are brighter than both objects and background, it is made sure the three blobs correspond to the fingers.

To locate the finger*tips*, the centers of gravity $\boldsymbol{C}_i, i = 1, 2, 3$ and the second moments of the blobs are calculated. Starting from the $\boldsymbol{C}_i$, pixels are tested along the major axis towards the image center until the first non-blob pixel is encountered, thus fingertip i is reached. Of course, this procedure is highly special purpose but justified by the computational speed. It could be easily replaced by other localization methods as e.g. Eigenspace projection ("Eigenfingertips"). However, the localization is not the main topic of this paper.

3.2 Windows for Classification

Four classifications have to be carried out to get the full information on the scene under the robot hand: VPL1 classifies a 65×65 window of I_1 centered between the fingertips at $\boldsymbol{C} = 1/3 \sum_i \boldsymbol{C}_i$. The result is the identity and position of the object.

VPL2 classifies three windows of size 45×45 from I_1 independently of each other, which are centered at the fingertips. The size is chosen such that a part

of the finger is included as well as a part of the object. Thus the window provides information on the fingertip position, the contacted part of the object, the object position and the relative fingertip-object position. This is the basis for a judgement on the type of "grasping situation" by the classifier. The choice of a set of such prototypical grasping situations is crucial for the performance (section 4).

3.3 VPL-Classifier

The classifier is a trainable system based on neural networks which performs a mapping $\boldsymbol{x} \rightarrow \boldsymbol{y}, \boldsymbol{x} \in \mathbb{R}^M, \boldsymbol{y} \in \mathbb{R}^N$. In this case, the input dimension M is the number of pixels of the windows. The window vector $\boldsymbol{x}$ has to be mapped to a discrete valued output k that denotes the class. There is one separate output channel for each of the N output classes. Training is performed with hand-labeled sample windows $\boldsymbol{x}_i^{Tr}$ and binary output vectors $y_i^{Tr} = \delta_{ij}, i = 1 \ldots N$, coding the class $1 \leq j \leq N$. Classification of unknown windows $\boldsymbol{x}$ is carried out by taking the class k of the channel with maximal output: $k = \arg\max_i(\boldsymbol{y}_i(\boldsymbol{x}))$.

The VPL-classifier combines visual feature extraction and classification. It consists of three processing stages which perform a local principal component analysis (PCA) as feature extraction followed by a classification by neural expert networks. Local PCA can be viewed as a nonlinear extension of simple, global PCA [13]. "VPL" stands for the three stages: **V**ector quantization, **P**CA and **L**LM-network. The vector quantization is carried out on the raw image windows to provide a first data partitioning with N_V reference vectors $\boldsymbol{r}_i \in \mathbb{R}^M, i = 1 \ldots N_V$. For vector quantization the Activity Equalization Algorithm is used, which is an extension of the well-known "winner takes all" method that prevents node under-utilization by monitoring the average node activities. [3].

To each reference vector $\boldsymbol{r}_i$ a single layer feed forward network for the successive calculation of the principal components (PCs) as proposed by Sanger [12] is attached which projects the input $\boldsymbol{x}$ to the $N_P < M$ PCs with the largest eigenvalues: $\boldsymbol{x} \rightarrow \boldsymbol{p}_l(\boldsymbol{x}) \in \mathbb{R}^{N_P}, l = 1 \ldots N_V$. To each of the N_V different PCA-nets one expert neural classifier is attached which is of the Local Linear Map – type (LLM-network), see e.g. [4] for details. It does the final mapping $\boldsymbol{p}_l(\boldsymbol{x}) \rightarrow \boldsymbol{y} \in \mathbb{R}^N$. The LLM-network is related to the self-organizing map [8]. It can be trained to approximate a nonlinear function by a set of locally valid linear mappings.

The three processing stages are trained successively, first vector quantization and PCA-nets (unsupervised), finally the LLM-nets (supervised). Classification of input $\boldsymbol{x}$ is carried out by finding the best match reference vector $\boldsymbol{r}_{n(\boldsymbol{x})}$, then mapping $\boldsymbol{x} \rightarrow \boldsymbol{p}_{n(\boldsymbol{x})}(\boldsymbol{x})$ by the attached PCA-net and finally mapping $\boldsymbol{p}_{n(\boldsymbol{x})}(\boldsymbol{x}) \rightarrow \boldsymbol{y}$.

The major advantage of the VPL-classifier is its ability to form many highly specific feature detectors (the $N_V \cdot N_P$ local PCs), but needing to apply only $N_V + N_P - 1$ filter operations per classification, thus large object domains can be represented and detected without the need of many (and computationally expensive) filter operations. The VPL-classifier has been successfully applied

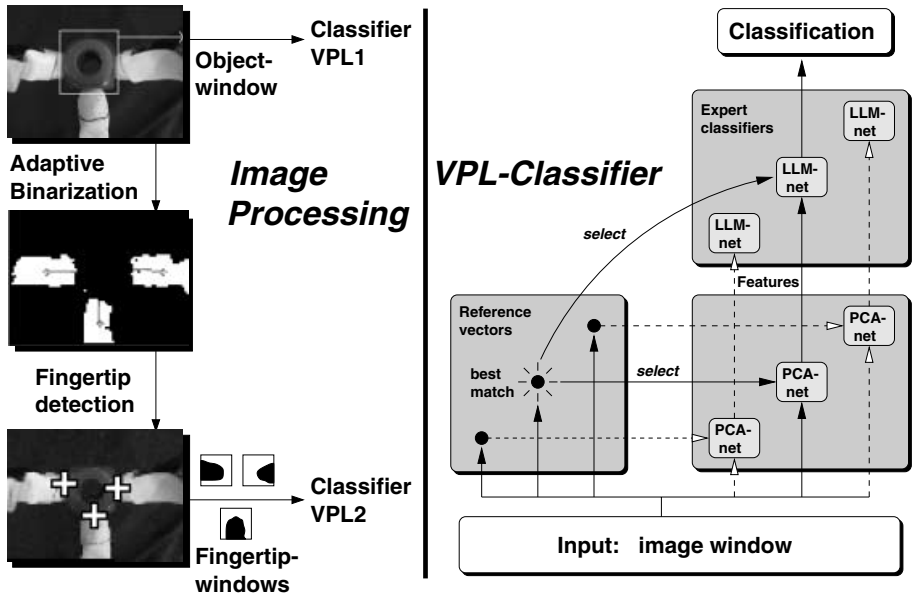

Fig. 3. System architecture. Left: From the input image the regions of the fingers can be easily segmented and the fingertips located (section 3.1). Around each fingertip a window is cut out of the input image (section 3.2). The three windows are fed to the VPL-classifier which classifies the "grasping situation" for each fingertip separately (section 3.3).

to several vision tasks, for details see [4]. Especially it could be shown that classification performance and generalization properties are well-behaved when the main parameters, i.e. N_V, N_P and the number of nodes in the expert LLM-nets N_L are changed.

4 Results

VPL1 classifies the N_O different objects and additionally up to N_Q qualitatively different classes for the position, so the VPL output dimension is $N = N_O + N_Q$. Fig. 4 shows typical grasping situations in which the object position matters most: The first two situations (from left to right) are stable (classes 1,2). The third is unstable though all fingers touch the object because the object can fall out of the grasp to the right (class 3). The fourth is an unstable version of the first (object tilted, class 4). So there are four position classes for this object. VPL2 needs not to be evaluated.

By contrast, Fig. 2 shows an object where the stability of the grasp cannot be easily derived from the object position. Hence, after recognition of the object type by VPL1, the "grasping situation" on the fingertips has to be judged by VPL2. The left and right fingers are in a correct position in all three situations,

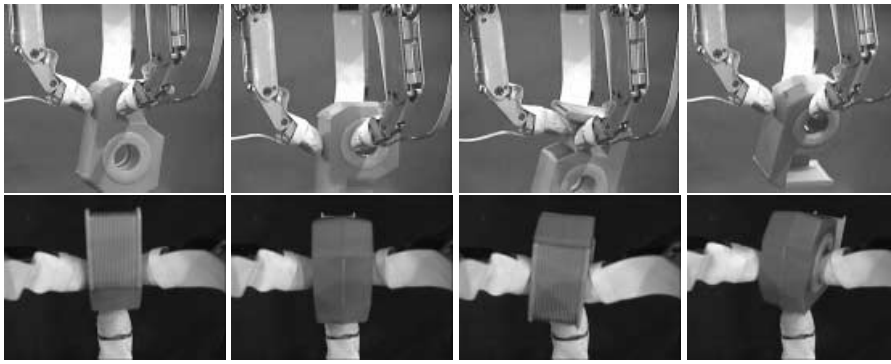

Fig. 4. Typical grasping situations in which the object position is decisive for grasp stability. Above view from the side, below the same situations as seen by the hand camera. From left to right: (1,2) stable position, (3) unstable though all fingers touch the object because the object can fall out of the grasp to the left, (4) unstable version of the (1), since object is tilted.

but the middle finger is lose in the second and third scene due to too strong bending. Note this causes a noticeable tilt of the object only in the third situation, not in the second (in the view of the hand camera!). So this object needs two classes for VPL2: finger correct / above.

We used in total $N_O^{VPL1} = 5$ objects with up to $N_P^{VPL1} = 4$ position classes for VPL1. Note the definitions of the position classes differ from object to object, also there may be fewer classes than N_P^{VPL1}. The architecture could be refined by using one VPL to classify the object type and then select another VPL specialized to the position classes of this one object. But the current scenario is simple enough to be covered by a single VPL1. For VPL2, $N^{VPL2} = 5$ classes were used to judge the position of the fingertips relative to the objects: "correct", "below", "above", "right" and "left".

The system was tested on a series of 60 images using the "leaving one out" method. For VPL1, $N_V = 3, N_P = 6, N_L = 9$ proved to be sufficient to reach a rate of 96% correct classifications. For VPL2, with $N_V = 4, N_P = 8, N_L = 20$ 88% correct classifications could be reached. The slightly larger nets (as compared to VPL1) and the lower classification rate reflect the greater number of degrees of freedom since the views of the fingertips are from different directions. Moreover, there are partial occlusions and overshadowing.

5 Discussion and Acknowledgement

A system for the qualitative visual judgement on the success or failure of a grasping action of a robot hand was presented. As the judgement has to provide information that can be used for further robotic actions, a set of "grasping situations" was defined for each object. Still, the number of objects and grasping situations is small. As we will get more images and more complex scenes, the

architecture will have to be enlarged by a single object classifier which activates specific object-position– and fingertip-position– classifiers individual for each object. Moreover, in the future sensor fusion between visual and oil pressure data will have to be developed.

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) within research Project SFB 360 "Situated Artificial Communicators".

References

1. AMP Incorporated, Valley Forge, PA 19482. *Piezo Film Sensors Technical Manual*, 1993.
2. G. Canepa, M. Campanella, and D. de Rossi. Slip detection by a tactile neural network. In *Proc. ICIROS 94*, volume 1, pages 224–231, 1994.
3. G. Heidemann and H.J. Ritter. Efficient Vector Quantization using the WTA-rule with Activity Equalization. *Neural Processing Letters*, 13(1):17–30, 2001.
4. G. Heidemann. *Ein flexibel einsetzbares Objekterkennungssystem auf der Basis neuronaler Netze*. PhD thesis, Univ. Bielefeld, Techn. Fak., 1998. Infix, DISKI 190.
5. R.D. Howe and M.R. Cutkosky. Dynamic tactile sensing: Perception of fine surface features with stress rate sensing. *IEEE Transactions on Robotics and Automation*, 9(2):140–150, 1993.
6. Interlink Electronics, Europe, Echternach, G.D. de Luxemburg. *The force sensing resistor*, 1990.
7. Ján Jockusch. *Exploration based on Neural Networks with Applications in Manipulator Control*. PhD thesis, Universität Bielefeld, Technische Fakultät, 2000.
8. T. Kohonen. Self-organization and associative memory. In *Springer Series in Information Sciences 8*. Springer-Verlag Heidelberg, 1984.
9. H. Liu, P. Meusel, and G. Hirzinger. A tactile sensing system for the DLR three-finger robot hand. In *Proc. ISMCR 95*, pages 91–96, 1995.
10. R. Menzel, K. Woelfl, and F. Pfeiffer. The development of a hydraulic hand. In *2nd Conf. on Mechatronics and Robotics*, pages 225–238, 1993.
11. Robert Rae. *Gestikbasierte Mensch-Maschine-Kommunikation auf der Grundlage visueller Aufmerksamkeit und Adaptivität*. PhD thesis, Universität Bielefeld, Technische Fakultät, 2000.
12. T.D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.
13. Michael E. Tipping and Christopher M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, February 15 1999.
14. M.E. Tremblay and M.R. Cutkosky. Estimating friction using incipient slip sensing during a manipulation task. In *Proc. ICRA 93*, volume 1, pages 429–434, 1993.