A Vision Based Robot System for Arranging Technical Objects

Stefan Kunze, Dr. Josef Pauli
Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Preusserstr. 1-9, D-24105 Kiel, Germany

97RO047

Abstract

Robot programming by demonstration simplifies the task of robot programming. Our implemented vision based robot system makes use of this approach and is able to arrange objects in a 2D-scene (e.g. a conveyor belt). To perform this task, the video camera takes images from two relevant objects and the movement of the robot hand is determined in such a way that both objects are arranged in a desired manner. The taught relation between two objects, e.g. a screw-wrench at a screw-nut, can be restored automatically, independent of their initial position and orientation. Almost all necessary information is extracted from images of the scene (very little a priori knowledge). The procedure of object recognition is based on the contour of the objects and derived features. The recognition procedure is invariant w.r.t. scaling, rotation and position of the objects, and actually this implies the generalization ability.

1 Introduction

The subject of this project was to implement a vision based system which gives a robot the ability to rearrange objects in a desired manner. It puts into practice the method called *programming by demonstration* [3], [6]. Systems based on this method differ by the ability of performing their tasks in a generalized way. Beside the ability of extracting the contour and derived features which are necessary to recognize, localize and identify objects automatically, our system stores the spatial relationship between objects. This relation can be restored by the robot in the sense that it is independent of the initial position and orientation.
There are some premises. Objects are assumed to be flat so there is no need of a three dimensional reconstruction of the scene. The pair of objects which shall be arranged must not overlap mutually or with other objects in the scene, so that the whole closed contours of the desired objects can be extracted. The object which shall be manipulated has to be grasped. This separate task is complicated and will not be part of this paper. However the object which is grasped is recognized automatically.

The system consists of four main parts:

1. *Modelling of shapes and objects*
   Images of the objects are taken which shall be arranged (appearance based object recognition [8]). The required models are extracted and stored.

2. *Calibration*
   The calibration takes place by multiple positioning of an object in the scene and relies on the localization procedure. This part will not be discussed in this paper.

3. *Demonstration of the spatial relation*
   By positioning a *grasped object*, it is related geometrically to another object, the *target object*, on the working plane. The relation between these two objects is stored afterwards.

4. *Arrangement of objects*
Known and unknown objects in the scene which can be arbitrarily positioned and orientated are automatically selected, recognized and localized. The grasped object is positioned by the robot so that the taught relation is restored.

2 Configuration of robot and camera

The STÄUBLI RX90 robot is characterized by six rotation joints so that the end effector can be arbitrarily positioned and orientated in the working space. The processing unit solves the problems of inverse kinematics and inverse dynamics. In essence only the position and orientation of the working tool must be sent to the processing unit. The images are taken by one grey scale camera which is mounted fix on a tripod. The size of an image is 512×512 pixels. The focal length of the lense was 12 millimeters. The optical axis and the object plane are perpendicular to each other. Image processing and object recognition was performed on a Sun SPARC workstation.

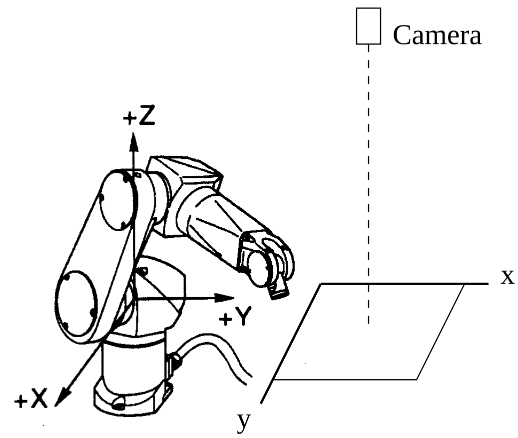

Figure 1: Configuration of robot and camera

3 Object recognition

3.1 General requirements

The choice of the method which recognizes, localizes and selects objects is determined by the properties of the images of the scene. First, the vision system must be able to distinguish between parts of the robot and objects in the image. A direct identification of robot parts in the image is difficult because of their nonrigidity. Second, it must be able to separate the image background from the desired objects. Because of the nonhomogenous background and noise there can be contours to which no real object belongs to. Third, it must also be able to distinguish between the objects themselves. The scene can be enriched with objects which are not intended to be arranged, these objects must be rejected. Finally, the most important point is that the recognition procedure has to cope with the arbitrary position, orientation and scaling of the objects.
To fulfil these requirements, we use a model based approach. Therefore, the images of the desired objects are taken in the modelling phase and suitable models are computed through image processing. Thus the problem of assigning an initial meaning [4] is solved. During the recognition procedure only objects to which a model fits are regarded. In this pragmatic way the system is informed about relevant objects for further recognition.

3.2 Representation of the contour and matching procedure

The discussion in the previous chapter has shown the necessity of a sophisticated representation of the contours and a suitable matching method. Contour based approaches have been proved to be favourable, e.g. in [7], [5] and [2], which in particular differ in their complexity.
Arkin et al. [1] published an efficient method for comparing polygonal shapes. They establish the notion of the *turning function* which represents the shape of an object. In the case of piecewise constant turning functions they present an $O(mn\log(mn))$ matching algorithm, where m

and n are the numbers of vertices of the two polygons. This algorithm has been applied and turned out to be very fast.

Let Q be a point on the contour of the object O with the length l_O. If one walks from Q to a point P on the contour so that the interior of the object is on the left side, one has covered a way of length $w \in [0, l_O]$. At first, one assigns to w the angle in radians between the tangent in P and the horizontal, in consideration of the circulations. Through a parametrization $[0,1] \to [0, l_O]; s \mapsto s \cdot l_O$ one obtains the turning function $\Theta_{A,Q} : \mathbb{R} \to \mathbb{R}$, where A is the shape of object O. The turning function is easily extended from the unit interval to $\mathbb{R}$ by including the circulations, i.e. by adding or subtracting multiplies of 2π.

Figure 2 shows an example of a turning function. The parametrization implies the invariance under scaling. There are two degrees of freedom, the choice of the starting point Q and the orientation of the object. This leads to the equivalence class

$$[\Theta_A] := \{\Theta_{A',Q'} \mid \exists\, t, \varphi \in \mathbb{R}\, \forall s \in [0,1] : \Theta_{A',Q'}(s+t) + \varphi = \Theta_{A,Q}(s)\} \tag{1}$$

of turning functions which belong to the same shape A.
The metric on these equivalence classes is defined as

$$d_2(A,B) = \left(\min_{t \in [0,1], \varphi \in \mathbb{R}} D_2^{A,B}(t,\varphi) \right)^{\frac{1}{2}} \quad \text{where} \quad D_2^{A,B}(t,\varphi) = \int_0^1 |\Theta_A(s+t) + \varphi - \Theta_B(s)|^2\, ds \tag{2}$$

It is shown that

$$d_2(A,B) = \left(\min_{t \in [0,1]} [h(t) - (\alpha - 2\pi t)^2] \right)^2 \tag{3}$$

where

$$h(t) = \int_0^1 [\Theta_A(s+t) - \Theta_B(s)]^2\, ds \quad \text{and} \quad \alpha = \int_0^1 \Theta_B(s)\, ds - \int_0^1 \Theta_A(s)\, ds. \tag{4}$$

In the case of an approximation of the contour by straight lines the turning function becomes a piecewise constant function. This leads to a simple representation of the form

$$\widetilde{\Theta}_{A,Q_A} = \{(\sigma_0^A, \alpha_0^A), \cdots, (\sigma_{m_A-1}^A, \alpha_{m_A-1}^A)\} \qquad \text{with} \quad m_A = n_A + 1, \tag{5}$$

where n_A is the number of vertices of the polygon which approximates the contour of a shape A, σ_i^A is the i-th supporting point and α_i^A the angle between the horizontal line and the tangent in the current point P in the environment on the right side of σ_i^A. The algorithm in [1] which computes the distance between two shapes A and B is of complexity $O(m_A m_B \log_2(m_A m_B))$. A side effect of the algorithm is that the best matching angle between the two objects is computed.

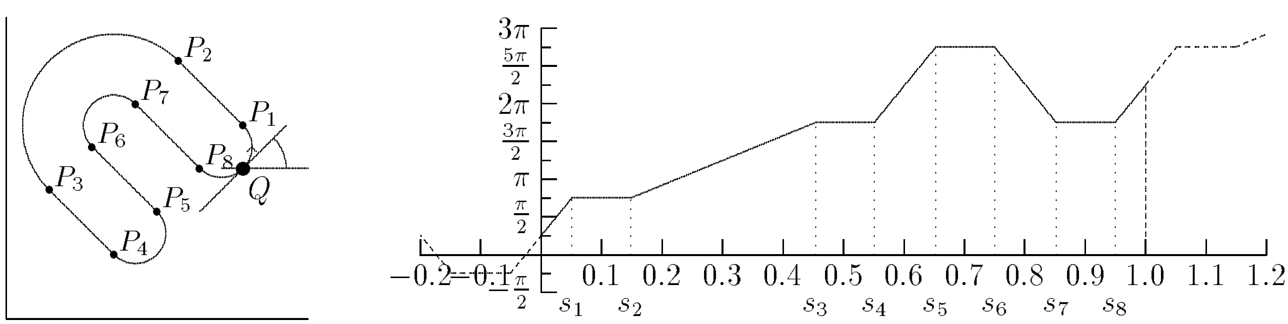

Figure 2: Object and its turning function

3.3 Modelling of objects

The shape is represented by the piecewise constant turning function and the normalized area $F_A := \frac{F_O}{l_O^2}$. F_O is the area and l_O the contour length of the image object. F_A is a measure for the circularity of the shape. The circularity of a narrow object is small and a circle shaped object has maximum circularity. F_A is used as a preselection criterion to decide whether two shapes are similar enough. So the algorithm which computes the distance between two turning functions must only be applied to shapes with similar circularity and this makes the whole matching procedure more efficient. Note that the circularity of two shapes can be equal for two different shapes, e.g. if the two shapes are mirrored. So the comparison of the circularities is only a necessary but not sufficient criterion.
The object is modelled by its shape and features that determine its size, i.e. its area F_O and its contour length l_O. F_O and l_O serve as selection criterion for objects.
We have extentend the metric on shapes to a metric on objects in terms of

$$d_O(O_1, O_2) = \sqrt{d_2(A_1, A_2)^2 + [\omega \cdot (|\frac{F_{O_1}}{F_{O_2}} - 1| + |\frac{F_{O_2}}{F_{O_1}} - 1|)]^2}. \tag{6}$$

The parameter $\omega \in \mathbb{R}_{>0}$ weights the relative errors of the areas. If ω tends to zero, only shape is measured.

To decide whether two objects are similar enough the distance must be compared with a threshold. If the distance of two objects is lower than this threshold, the objects are accepted as similar, otherwise they do not match.
This method is applied on all objects in the image. The obtained set of objects includes the candidates of target objects and the object which is grasped. The grasped object is determined by the method described in the next section.

4 Recognition of the grasped object

The appearance of the gripper distuinguishs itself from the objects during the grasping process. The gripper moves while the other objects are fix in the image. This feature is used to determine which of the recognized object is the grasped one. Two images are taken while the gripper is closing. By applying the difference operator on these images, one get an image with high grey values in the environment of the end effector. Additionally, edges are extracted to get dominant points from which the convex hull is computed. This convex hull is the geometric description of the gripper range. The object which intersects with the convex hull or lies inside the hull is the

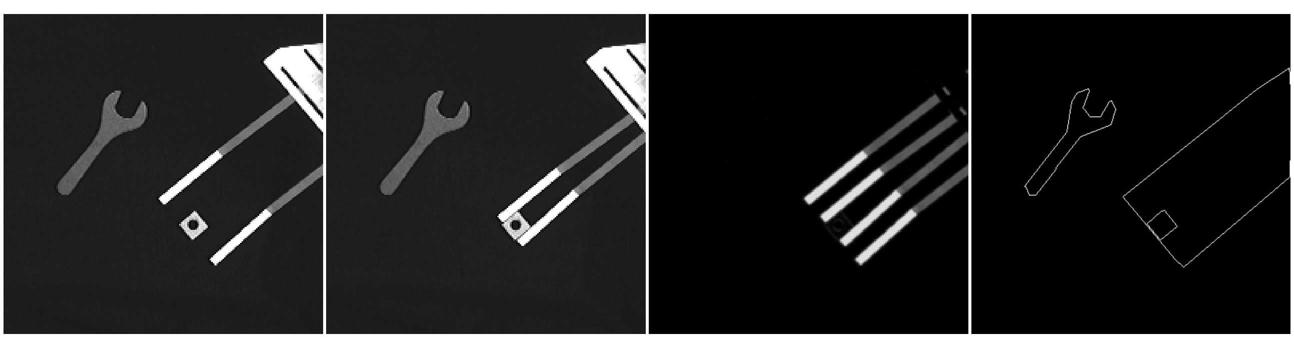

Figure 3: Image of the scene with opened gripper, image of the scene with closed gripper, difference image, contour of the grasped object and the target object with the convex hull which describes the gripper range.

grasped object. This method has the advantage that it is fast and in particular independent of the appearance of the effectors. Figure 3 illustrates this method.
According to this approach of object recognition, the system is able to recognize the candidates of target objects and the grasped object.

5 Learning and application of the knowledge

The operator puts two objects in the working plane and uses the control panel of the robot to arrrange the two objects manually and thus demonstrates the system the desired geometrical relation between both. The vision system computes the contours, recognizes the objects and computes a tripel which represents the relation beween the grasped and the target object. The tripel is defined by

$$(O_g, R, O_t), \tag{7}$$

where O_g und O_t are the models of the grasped resp. the target object. R is the tripel (ϕ, z, v) that stores the data to rearrange the two objects. ϕ is the angle in radians the grasped object was rotated. To allow that two objects are rearranged upon each other, z stores the vertical offset. $v = (x, y)$ describes the scale invariant position of the common center of area of the two objects.
Figure 4 shows an example of two related objects. The triangular shaped block was positioned side by side to the halfcircle shaped box in the way that two corners meet. An arbitrary number of relationships can be learned in this way for further application. The knowledge is represented by the union of tripels in (7).

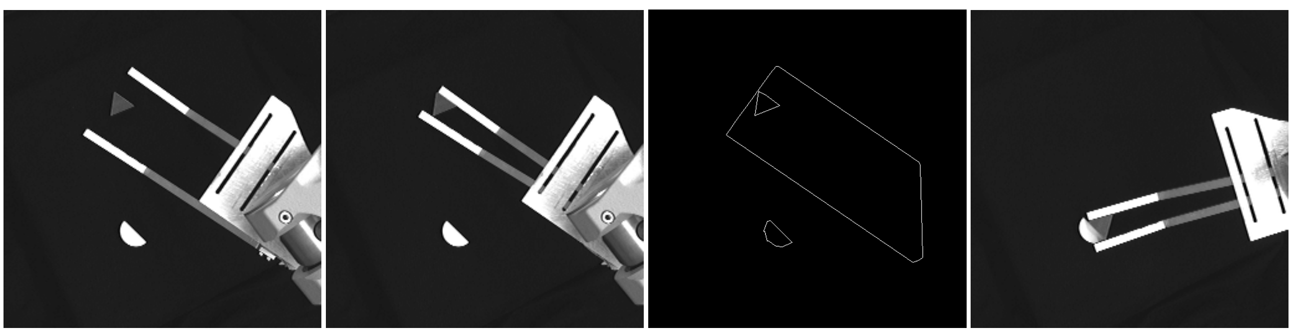

Figure 4: Scene with open gripper and scene with closed gripper before the positioning of the object by the operator, the contour of both objects with the convex hull which describes the range of the gripper, scene after relating the objects through the programmer.

The relation learned in the demonstration phase is the basis for rearranging the objects automatically. The objects in the relation set serve as the models to recognize objects in the image of the scene taken by the camera. Pairs (O_g, O_t) of candidates are extracted from the image. If there exists a matching tripel (O_g^r, R, O_t^r) in the stored relation tripels, the relative orientations of O_g and O_t are computed from O_g^r and O_t^r and thus the object O_g can be manipulated by the robot so that the relation code in R is restored. Therefore, the distance of a pair of objects is measured by

$$d_R((O_{g_1}, \cdot, O_{t_1}), (O_{g_2}, \cdot, O_{t_2})) = \max(d_O(O_{g_1}, O_{g_2}), d_O(O_{t_1}, O_{t_2})). \tag{8}$$

The metric on triples of the relation is in terms of

$$d_R((O_g, O_t), (O_g^*, O_t^*)) = \min\{d_R((O_g, O_t), (O_g^r, O_t^r)) \,|\, (O_g^r, \cdot, O_t^r) \in R\}. \tag{9}$$

A threshold operator is used again to decide whether two triples match or not.
Figure 5 demonstrates the precision the objects are rearranged. The underlying relation was the

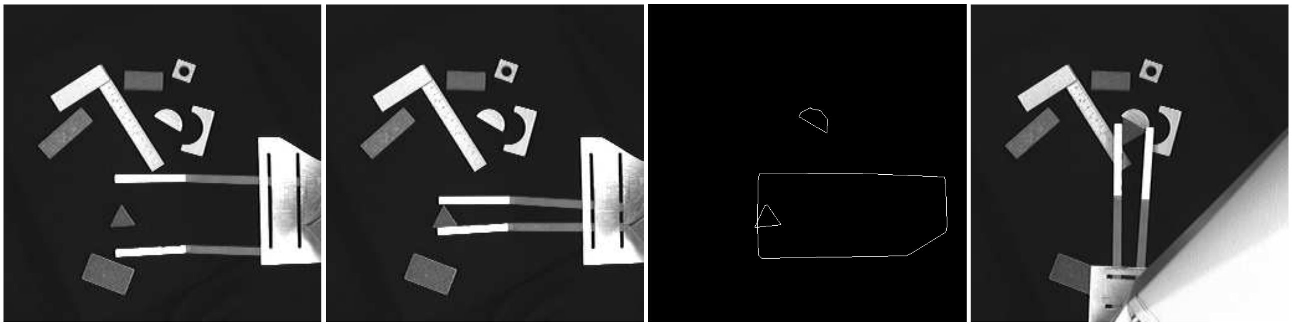

Figure 5: The restored relation of figure 4.

relationship between the triangle shaped block and the halfcircle shaped block. Consequently they are recognized as grasped object resp. target object and are manipulated by the robot in the desired manner.

6 Conclusions

We developed a system to arrange objects which works even if the objects are translated and rotated. The method of *programming by demonstration* is applied by using vision based object recognition. The theoretical basis for the shape matching procedure was given by [1]. A model based matching procedure for object recognition and localization was derived. The starting point of this procedure is a symbolic contour description which is generated by simple preprocessing which has to be adapted to the environment. Despite of unavoidable shadows and approximation of the contours by straight edges the programmed relations were restored with high precision. The recognition method is determined only in the two-dimensional plane, nevertheless many applications are conceivable, especially in automotive applications.

References

[1] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, J. S. B. Mitchel, *An efficiently computable metric for comparing polygonal shapes,* PAMI, Vol. 13, No. 3, pp 209-215, 1991.

[2] Alejandro Blumenkraans, *Two-dimensional object recognition using a two-dimensional polar transform,* Pattern Recognition, Vol. 24, No. 9, pp 879-890, 1991.

[3] S. Bocionek, M. Sassin, *Lernen durch Vormachen,* Informatik-Spektrum, Band 17, Heft 5, pp 309-311, 1994.

[4] G. Cottrell, B. Bartell, C. Haupt, *Grounding Meaning in Perception,* in H. Marburger, editor. 14th German Workshop on Artificial Intelligence, Fachberichte 251, pp. 307-321, 1990.

[5] Susan R. Dubois, Filson H. Glanz, *An autoregressive model approach to two-dimensional shape classification,* PAMI, Vol. 8, No. 1, pp 55-66, 1986.

[6] H. Friedrich, S. Münch, R. Dillmann, S. Bocionek, S. Sassin, *Robot Programming by Demonstration (RPD): Supporting the induction by Human Interaction,* Machine Learning, 23, pp. 163-189, 1996.

[7] P. Johansen (Hrsg.), *Dynamic Programming solves approximate planar matching,* Theory & applications of image analysis, 7th Scandinavian Conference on Image Analysis 1991, pp 99-109, World Scientific, 1992.

[8] H. Murase, S. Nayar, *Visual Learning and Recognition of 3D Objects from Appearance,* International Journal of Computer Vision, 14, pp. 5-24, 1995.