Annales Univ. Sci. Budapest., Sect. Comp. 17 (1998) 405-416

RECOGNITION BASED OBJECT CLASSIFYING SYSTEM IN ROBOT ENVIRONMENT

Z. Vámossy, L. Csink (Budapest, Hungary)
J. Schröder, N. Okulan (Bremen, Germany)
I. Katzer, F. Molnár, E. Szabó (Budapest, Hungary)

Abstract. At various labour processes in a robot environment, such as conveyance of materials, classification, mounting or quality control, the recognition of individual pieces of work is imperative these days. Several image processing based systems have been developed aiming at the solution of the above tasks. It is a general drawback of these systems, however, that in order to process, store and classify the objects fast, special hardware requirements are set and therefore these systems are usually not economical. Considering the above criteria in the framework of two projects, we have developed an experimental prototype of a PC-based robot vision system. The system capable of recognizing the various already known, taught shapes or completely unknown objects coming on a conveyor belt, automatically examining if the object is grippable, and co-operating with a BOSCH SCARA SR60-E robot, moving the objects into different storage units, depending on the result of the recognition.

1. Introduction

Our PC-based pattern recognition system co-operates with a BOSCH SCARA SR60-E robot. The objects are transported to the vision area of the robot by a conveyor belt. When the object has reached the end of the belt,

---

The developed working system can be seen at the Institute of Informatics at Kandó Polytechnic of Technology, Budapest after registrating by e:mail at Z.Vámossy, research group leader.

this fact is detected by a retro-reflective opto gate and the belt is stopped. Receiving the signal of the optical sensor, the robot arm moves over the belt ("Viewposition"), so the CCD-camera attached to the robotic arm views the object (see Fig. 1., Fig. 6). The information provided by the camera is converted to a grayscale image by a PC-video-digitizing card, which serves as the input to the image processing system. The image is 512*512 pels. The identification of the objects is performed based on this image. Our system does not deal with the height of the objects, so we suppose that the objects have about uniform height and they are not higher than the length of the gripper fingers.

The *first project* we describe was developed in co-operation of our Institute and Hochschule Bremen (Vámossy, Z., Schröder, J., Okulan, N.) - aimed at the fast *identification of simple objects* and also *their moving into target position*. The PC-based image processing system starts from the 2D image containing rectangles, after having performed binarization and noise removal, and determines the main features: area, perimeter, nodes, edges, centre of gravity, etc. A new method was worked out to find the nodes ("half-method", "quarter method") that converges fast and correctly. The system is capable to differentiate between brick shaped objects. The pattern recognition program classifies the detected objects, and determines the grabbing position and orientation for the robot, or sends a signal that the object cannot be grabbed.

In the *second subsequent project* (Kandó Polytechnic: Vámossy Z., Csink L., Katzer I., Molnár F., Szabó E.), we used the hardware configuration, which had been implemented in the first project, but we developed further the capability of the recognition part of the robot vision system, such as the image may contain several objects of arbitrary shape. The system is capable to work under varying light intensity. *The system* has been designed in a way that by a suitable setting of parameters it *is capable to differentiate between very similar, any shaped objects* as well. So the experimental robotized classifying system has general character, the set of the objects depends on the parameters of the applied robot gripper, the considered objects can be identified from their 2D binarised images. The system is not able to handle nested objects or distinguish between identically shaped objects with different textures.

2. Pattern recognition based classification of brick shaped objects in a robot environment

The objective of the system developed in the framework of the first project is the recognition and classification of brick shaped objects in a modular way that makes the further development of the system possible. Fig. 1 shows the connection of the hardware elements in a robot environment.

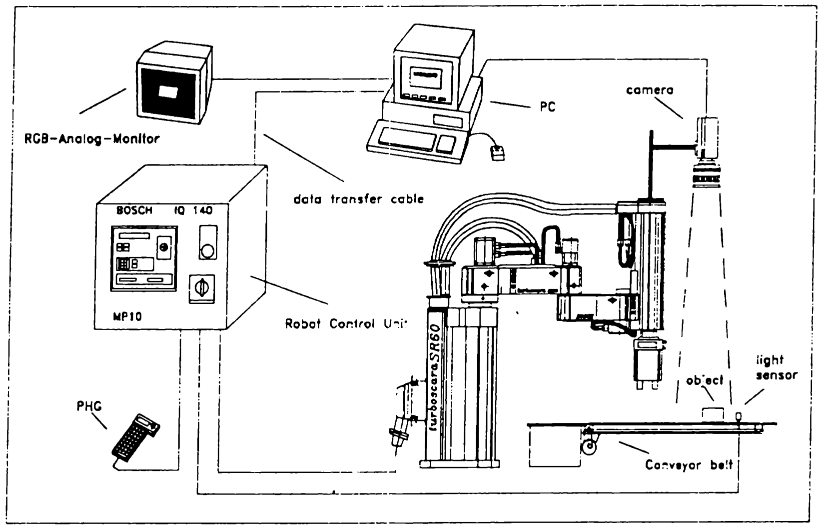

Fig. 1. Robotized classifying system

As the camera is fixed on the robot arm, it is easy to change the image scanning position of the camera (so called "Viewposition"), as well as the position of the conveyor belt in the working area. (This configuration can be advantageous in further projects when we try to track and grasp moving objects in the working space, for example objects travelling on the conveyor belt). The determination of the correlation between the co-ordinate system of the robot and the camera's image (calibration) is realized according to our previous work [1, 2]. The robot arm moves a brick shaped object with known parameters firstly in a so called synchronization position, which is interactively adjustable by the usage of the standard teachpendant of the robot system (PHG in Fig.1.). After taking a picture about the reference object the robot arm moves the object in X direction with a fix distance and the camera provides a newer image about the brick from the same "Viewposition". From these two images and the known parameters of the object the above relationship can be calculated by simple trigonometric functions and the real size of one pixel as well. The communication of the image processing computer and the robot control unit is provided by buffered data transfer through a sequential port. The interface makes it relatively easy to adapt an arbitrary control unit.

The image processing program is based on a traditional [5] binarisation and segmentation method. The 512*512 image is pre-processed by 4-neighbour averaging and median filtering, which make it more homogeneous and less noisy. Then, by a threshold parameter set by the user, the image is thresholded into a binary image.

The computing of the area, the centre of gravity and other features is similar to the methods used in [8]. The detection of corner points is, however, new. The "half method" and the "quarter method" (Fig.2) both compute the centre of gravity of a given part, and it iterates the computation between the newer and newer computed point and a selected node of the region as long as the distance between the computed points in two consecutive steps are closed enough. (In contrast to the quarter method that splits the sub-viewports each time in four parts, the half method splits the sub-viewports each time in two parts to find the new centre of gravity of the new sub-viewport.) The usage of one of the above methods depends on the orientation of the rectangle in the image. This method is fairly exact for rectangle shapes, so for objects with a slight difference it is suitable.

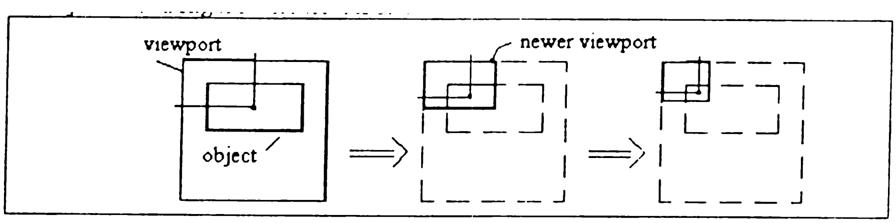

Fig. 2. "Quarter method" to find the corners of a rectangle

To realize the classification one needs - apart from the above features - the lengths of edges computed from the corner point co-ordinates, the perimeter and the radius, so that they were compared to the data earlier learnt and stored (knowledge based classification [5]).

3. Identification of arbitrarily shaped objects in varying illumination conditions

3.1. Image segmentation

3.1.1. Binarization automatically adapted to change of light conditions

As the system is based on optical processing, when illumination has a great importance. However, it is difficult to provide constant illumination, so the program must be tolerant to changes in illumination.

The following procedure is applied (similarly to [8]): the image is divided into small units (squares) in which the grayscale distribution is more homogeneous except for squares containing the contours of the objects. For each square we determined two grayscale values by which the spectre was limited. Collecting these local minima and maxima, we get a distribution that is easier to handle. Choosing a binarization threshold between the lowest maximum and the highest minimum, we get a suitable binary image.

3.1.2. Contour detection

Contour shape is uniquely described by chaincodes [5,7,8]. Starting from the actual point co-ordinates the next point can be determined by a coded direction according to the following table:

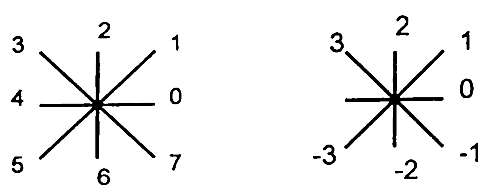

Fig.3. Direction codes and direction code differences

It is characteristic of a contour of a given object that it is closed, i.e. the starting and ending points coincide, moreover a minimal rectangle can be determined that contains the object.

To find a contour the program scans for the first foreground colour (from left to right at each line, and then downwards) which does not belong to any contour already found. If the program managed to find such a point, then the contour following algorithm is called, and afterwards this procedure is iterated. The following problems need to be solved:

- the object contour must be systematically followed, that is the directions compared to the previous one must be considered in a predefined order,
- it is not allowed to get into the inner part of the object,
- the points that cannot be attached to the chaincode must be handled (any point can appear in the chaincode only once). Small noises "hanging on" the contour must be deleted. Joining objects have to be separated.

3.1.3. Contour segmentation

Contour segmentation means that the contour is approximated by line segments [5] based on the chaincodes. From these points a preliminary point partition is made, from which the optimal partition is constructed by the split and merge algorithm.

The preliminary partition has to be such that if the chaincode remains the same for a set of points then they have to get to the same segment, while if the chain code changes a new segment starts. If we want to characterize the changes in the chaincode, we form the difference of the direction pointing to the contour point and the direction belonging to the contour point (Fig.3). Assigning these values to the contour points we get the difference chaincode list. This is not yet sufficient to characterize contour changes, as it is of a localized character. However, if some points before and after a contour point are taken and the differences are summed with a weight, then an approximately good curvature value can be computed. The sign of the curvature shows whether it is convex or concave.

The preliminary partitioning is based on the curvatures. Each curvature is either convex, or concave. At each point where the curvature changes from convex to concave or vice versa, a new segment starts. However, segments of length one unit are avoided.

3.1.4. Optimizing the segmentation and calculation of intersection points

The optimization of segmentation is performed by the split and merge algorithm [4]. The equation of the lines fitted on the segmentation points is computed by the regression method. When computing the regression lines, the starting point of the next segment is also considered, so that the segments should not separate, and the point of intersection of neighbouring regression lines should not differ from the contour with more than the maximum error. The optimized partition can be seen at Fig.4. *

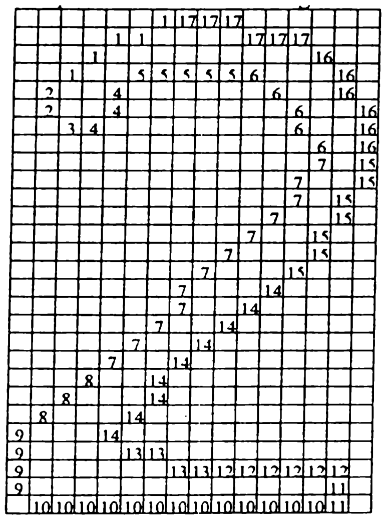

Fig.4. Optimized partition

3.2. Preparations for object manipulation

3.2.1. Finding the gripping order

As it is clearly important that no object should fall off the conveyor belt if it is in principle grippable, the one nearest to the end of the conveyor belt will have to be gripped first. If the result of the gripping check proves to be negative (i.e. the object cannot be gripped), then our system allows the movement of the belt further.

If the object were grippable, but a nearby object prevents this, then it is allowed to grip the preventing object first.

---

* The reason, why character images are presented as an example, is that these were used for testing the pattern recognition program on one hand, and even the smallest grippable object is too big to demonstrate on the other hand.

3.2.2. Examination of gripping possibilities

In order that a robot should be able to grasp an object, we need to find the edges opposite to each other. The edges must be nearly parallel and the distance between them must be in the motion range of the robot gripper. If there are several possible ways to grasp an object we need to provide priority for those cases when the opposite pairs of edges surround the centre of gravity, consequently there exists a common perpendicular to the edges that is not so far from the centre of gravity. In this case the robot can grasp and move the object in the target position in a secure manner.

The examination algorithm works with the approximating straight lines obtained by the segmentation. If the object is approached by three or less straight lines the examination rejects the possibility of grasping without any further check since secure gripping of those kinds of shapes is not given.

After examining whether the object has nearly parallel opposite edges, the next step is the evaluation of the collected pairs of edges or in other words the establishment of the fact which pair of edges is the most suitable for grasping. A priority order can be determined by using the following properties:

- The distance between the centre of gravity and the middle of the grasp line segment. The smaller this distance is, the smaller the momentum becomes, and so the grasping is more secure.
- The distance of the endpoints of the grasp line segment from the centres of the edges. Since if the edge is shorter than the width of the gripper, the grasping can more unstable.
- The third step is to select from the grippable pairs of edges the options that are approachable by the gripper without collision. Since the conveyor belt may consist of several objects at the same time, these objects can hinder each other in the grasping process. The check of approachability means that for the pair of edges with the greatest priority we check the environment of the target position and motion range of the gripper whether there is any other object there. If this necessary condition is not granted, we can reject the pair of edges as a possibility of grasping. This examination loop is iterated for opposite pairs of edges until the solution of the problem is found, or the priority value goes under a limit, or finally the last suitable pair of edges is checked without result.

3.3. Identification of the object

3.3.1. Calculation of the shape vector

The elements of the shape vector are the length-normalized values of the contour direction codes of the object. For that reason we have to calculate the lengths of the approximating line segments that are fitted to the contour segments and the sum of these lengths, which is the perimeter of the object (Fig. 5.a). Then the direction code differences of the neighbouring line segments are stored in an array (shape vector - Fig. 5.b).

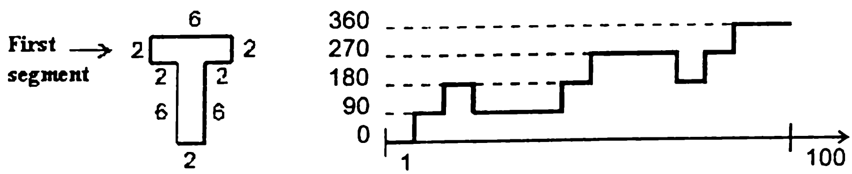

Fig. 5.a. and 5.b. The fitted segments and the shape vector's diagramm of the object like character "T"

3.3.2 Assessment of the "suggestion list"

The recognition of the target object is performed by its comparison to all reference objects. For each characteristic (area, perimeter, shape vector, etc.) an error factor is computed, then by weighting a resultant error. Summarizing the results in an ordered list (according to error value) we can start the evaluation of the "suggestion list".

For every error there exists a threshold value that must not be exceeded.

If several suitable reference objects exist, we select the one with the smallest resultant error.

However, it is important to point out the fact that a known object can also arrive on the conveyor belt rotated to its previous stored orientation, so the comparison has to be invariant for the rotation. Therefore the shape vector has to be rotated by every possible angle in the comparison process, and it has to be optimized for the square of errors.

$$HA = \min_{i=0}^{N-1} \frac{\sum_{j=0}^{N-1} ((A[j] - A[0]) - (Ar[(i+j) \bmod N] - Ar[i]))^2}{N},$$

where HA - the error of the shape factor,

A - the shape vector of the object to be recognized,

Ar - the shape vector of the reference object,

N - length of the shape vector.

4. Conclusions

The purpose of our vision based robot system, that is described in this paper, was to identify and grasp arbitrarily shaped objects in varying illumination conditions. The recognition process and the automatic examination if the object is grippable were implemented on a 486 DX4-100 PC and on a BOSCH SCARA SR60-E robot. The robot environment and the examined objects in the first project can be seen on Fig.6. A subset of the test elements for the second project is collected on Fig.7.

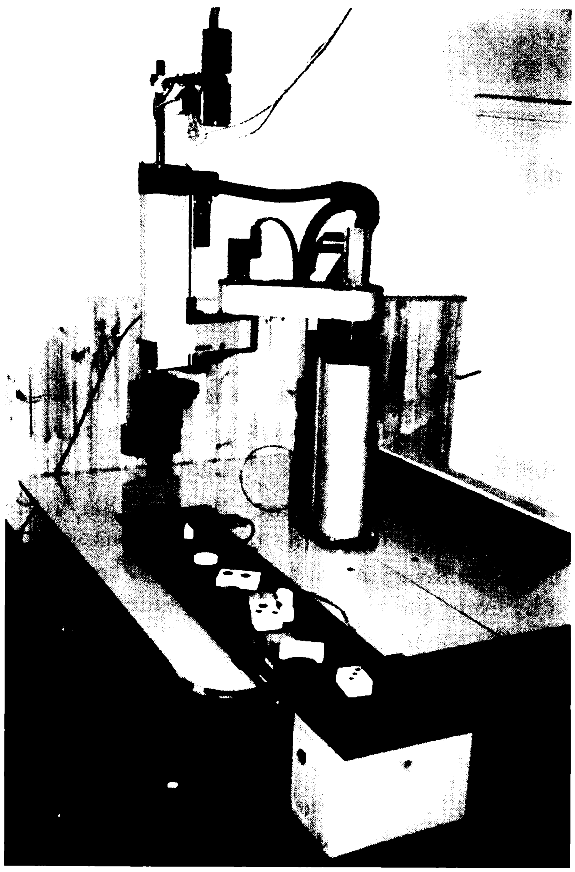

Fig.6. The robotised classifying system in the first project

The most important adjustable parameters are the followings: accuracy for the fitting of the regression elements; error threshold values for the recognition parameters, namely: area, perimeter, shape vector; minimal length of the object perimeter in pixel; the parameters of the parallel gripper, mainly the motion range of the fingers and the minimum distance between them.

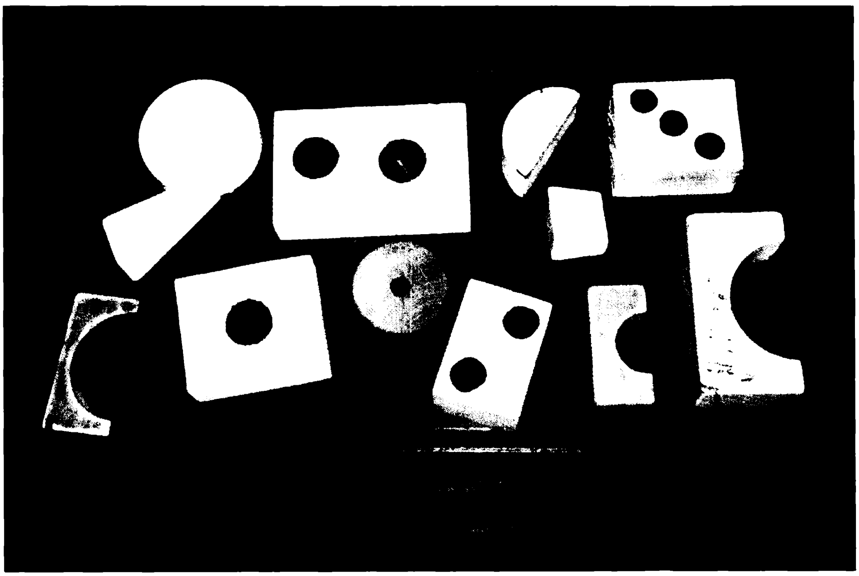

Fig.7. A subset of the test objects in the second project

The average cycle time from the beginning of an object's recognition until the robot grasps it is 1.54 second (without prelearned database). If we compare the parameters of the actual object to the items of the database then the duration of the process is about additionally 0.02 second for each item of the database elements. However, the average processing time on the PC side (checking by the TurboProfiler) is less than 0.6 second. The nearly 1 second difference comes from the communication time between the PC and the robot controller and the detection time of the signal of the optical gate.

Acknowledgement. The authors are grateful to Professor Sima, the director of the Institute of Informatics (Kandó Polytechnic of Technology) and Professor Greger, the director of Fachbereich Elektrotechnik (Hochschule Bremen) for their continuous encouragement and for establishing the necessary research conditions.

We acknowledge the support of the TEMPUS Office in Brussels (JEP 2309) for financing the purchase of BOSCH SCARA SR60-E robot.

References

[1] **Vámossy Z.**, *Industrial Robots-Introduction to Robot Technology,* technical report, 1993.

[2] **Vámossy Z.**, *Robotics I. Fundamentals of Robot Modelling, European Informatics Students Lecture Notes Series,* Budapest, 1994. (prepared in the framework of the Tempus project PROMISE).

[3] **Masuda I., Hagita N., Akiyama T., Takahashi T. and Naito S.**, Approach to Smart Document Reader System, *Proc. CVPR'85, San Francisco,* 1985, 550-557.

[4] **Niemann H.**, *Image Analysis and Understanding,* Springer, 1990.

[5] **Niemann H., Brünig H., Salzbrunn R. and Schröder S.**, A Knowledge-Based Vision System for Industrial Applications, *Machine Vision and Applications, Vol. 3,* Springer, 1990, 201-229.

[6] **Shimotsuji S.**, A Robust Drawing Recognition System Based on Contour Shape Analysis, *Proc. IEEE Conference, 1990,* 717-719.

[7] **Pavlidis T.**, *Algorithms for Graphics and Image Processing,* Springer, 1982.

[8] **Parker J.R.**, *Practical Computer Vision Using C,* John Whiley and Sons, New York, 1993.

Z. Vámossy, L. Csink
I. Katzer, F. Molnár, E. Szabó
Institute of Informatics
Kandó Polytechnic of Technology
P.O.B. 112.
H-1431 Budapest 8.
Hungary

J. Schröder, N. Okulan
Fachbereich Elektrotechnik
Hochschule Bremen
Neustadtswall 30
D-28199 Bremen
Germany