

Vision Based Robotic Interception in Industrial Manipulation Tasks

Ahmet Denker and Tuğrul Adıgüzel

Abstract—In this paper, a solution is presented for a robotic manipulation problem in industrial settings. The problem is sensing objects on a conveyor belt, identifying the target, planning and tracking an interception trajectory between end effector and the target. Such a problem could be formulated as combining object recognition, tracking and interception. For this purpose, we integrated a vision system to the manipulation system and employed tracking algorithms. The control approach is implemented on a real industrial manipulation setting, which consists of a conveyor belt, objects moving on it, a robotic manipulator, and a visual sensor above the conveyor. The trajectory for robotic interception at a rendezvous point on the conveyor belt is analytically calculated. Test results show that tracking the target along this trajectory results in interception and grabbing of the target object.

Keywords—robotics, robot vision, rendezvous planning, self organizing maps.

I. INTRODUCTION

VAST majority of applications for industrial purposes utilize robotic manipulators. Sensing a target object and reaching it for grasping could be considered as a common goal for manipulation tasks. The ability of sensing a moving object is important for the efficiency of manipulation tasks in these industrial settings.

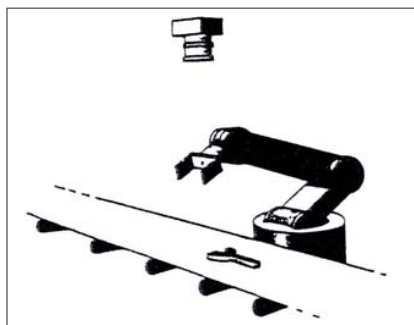


Fig. 1. Manipulation platform using vision system

Robots have had less impact in applications where the work environment and object placement cannot be accurately controlled. The limitation is mainly due to the lack of sensory capability. Using a visual set-up for robotic manipulation when

working with targets whose position is unknown, provides substantial advantages to solve these problems[1]. For this reason from early of seventies to recent days, efforts on combining vision and robotic applications have appeared in many studies[2,3,4,5,6]. A depiction of robotic manipulation platform utilizing visual system is shown in Fig.1.

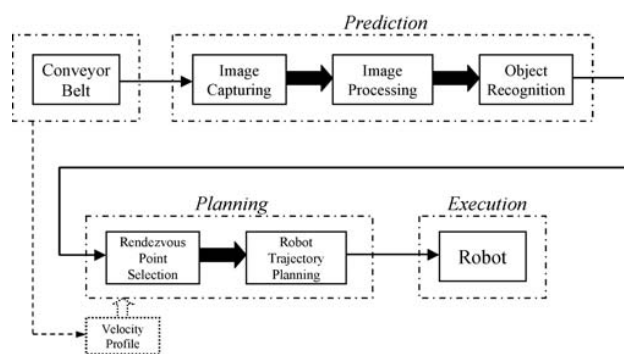


Fig. 2. Manipulation platform using vision system

The block scheme in Fig.2 summarizes the set-up which is implemented. There is a conveyor belt and a digital camera takes pictures of the region of interest. In the prediction block, the system gets the captured image data, analyzes it and decides if there is a target object moving on the conveyor. If the system detects a target object, planning of motion for the end effector, with respect to the target object's motion, is done in the next block. Reference trajectory for the end-effector to intercept the object is calculated at this stage. Executing the calculated trajectory on the robotic manipulator is the final step.

Although there is an integrity between the blocks of such a system, the main emphasis of this study is on the part of the system providing visual sensory capability and planning a robotic interception at a convenient rendezvous point. Then, the approach to the problem can be split in two main parts as robot vision and robotic interception.

In section 2, image processing part of the system is explained and the algorithms used for feature extraction and recognition are given. In section 3, motion planning for robotic interception is presented. Extraction of the initial positions of determined target objects, and calculation of the tracking trajectory are examined. The paper concludes with the simulation and experimental results.

Manuscript received February 02, 2006.

A. Denker is with the Electronics Engineering Department, Ankara University, Besevler, Ankara, 06100 Turkey (e-mail: denker@eng.ankara.edu.tr).

T. Adıgüzel is with the Electronics Engineering Department, Ankara University, Besevler, Ankara, 06100 Turkey (corresponding author to provide phone: +90-312-212 67 20 ext.1466; fax: +90-312-212 54 80; e-mail: adiguzel@eng.ankara.edu.tr).

II. OBJECT RECOGNITION

Vision part of the system is modeled as in the diagram shown in Fig.3. Scene covers the camera's field of view on the belt, and is lightened by using two distinct light sources which are located at opposite sides of the camera.

The inputs for this part are captured images, and generated outputs are symbolic definitions of the skeleton images. Symbolic definitions contain feature data of the objects. Obtaining these features could be considered in two steps: i) distinguishing objects from medium and obtaining a skeleton of the image; ii) determining object definitions by using pre-defined application data.

By using extracted feature, a classifier determines whether the seen object is a target or not.

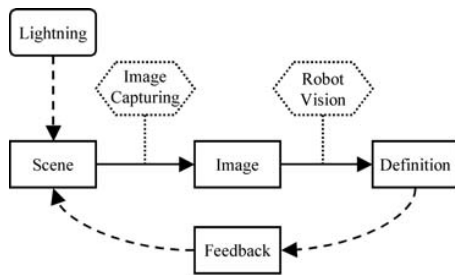


Fig. 3. Robot Vision Scheme

The object recognition method in this study, utilizes corner information, which is obtained by using an 8-directional chain code, as features of objects, and Kohonen's self organizing maps as classifiers[7]. The method operates in three steps; image processing, feature extraction and classifying.

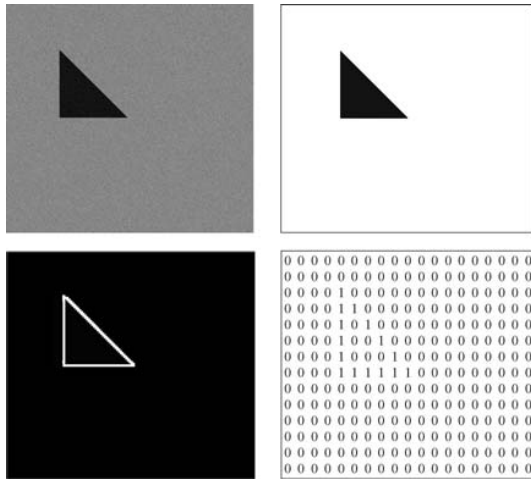


Fig. 4. Edge detection for a triangular object

A. Image processing

In image processing phase; a histogram based approach is used to detect the objects on the conveyor. By simply defining

a threshold level for histogram of the image, it is possible to determine whether there is an object or not. Once the object is detected, then gradient method is implemented to extract edges of the object. Following thresholding the frame to get a binary image, edges of the object are filtered by looking for the maksimums and minimums in the first derivative of the image[8]. An example edge detection process is shown in Fig.4.

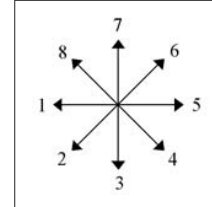


Fig. 5. 8-directional chain code

B. Feature Extraction

For extracting features of objects, edge information of each objects are encoded in accordance with an 8-directional chain code as shown in Fig.5. This code is obtained by determining the direction and number of 1's in a binary image. For instance, the chain code of a triangular object shown in Fig.4 is:

$$c = [3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 8 \ 8 \ 8 \ 8 \ 8 \ 8].$$

The chain code vector, $\mathbf{c} = [c_1, c_2, \dots, c_n]$, is subjected to two transformations to obtain a new vector \mathbf{d} , which carries information that is independent of rotation and size. The following two transformation algorithms are implemented to obtain \mathbf{d} from \mathbf{c} .

Algorithm 1: [First transformation]

```

for i=1 to n
    if  $c_i = c_{i+1}$  then  $c_{i+1} = 0$ 
    else  $c_{i+1} = c_{i+1} - c_i$ 
end
  
```

Algorithm 2: [Second transformation]

```

j=0, k=0
for i=1 to n
    if  $c_i \neq 0$  then  $k = k + c_i$ 
    if  $c_{i+1} \neq 0$  and  $k \neq 0$  then  $j = j + 1$ 
     $d_j = k$ 
     $k = 0$ 
end
  
```

Using \mathbf{c} and \mathbf{d} vectors, characteristic vectors \mathbf{v} with eight elements which are assigned to each object class as features, are obtained.

$$\mathbf{v} = [v_0, v_1, v_2, \dots, v_7]. \quad (1)$$

The elements of a characteristic vector have the following definitions.

- v_0 : number of corners,
- v_1 : ratio of the longest side to the shortest side, rounded off to integer,
- v_2 : number of angles between 0 and 45 degrees,
- v_3 : number of angles between 46 and 90 degrees,
- v_4 : number of angles between 91 and 135 degrees,
- v_5 : number of angles between 0 and -45 degrees,
- v_6 : number of angles between -46 and -90 degrees,
- v_7 : number of angles between -91 and -135 degrees.

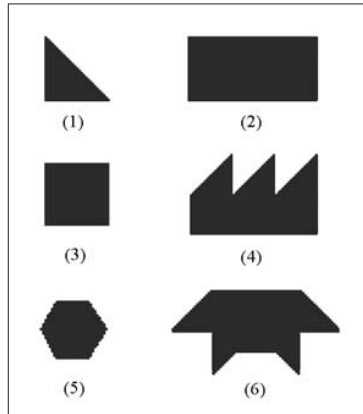


Fig. 6. Object classes to be recognized

C. Classification

The characteristic vectors obtained by the previously explained way are then used as inputs to the self organizing maps. The characteristic vectors, for the objects used for training in Fig.6, are in the following table.

TABLE I
CHARACTERISTIC VECTORS ASSIGNED TO OBJECTS

Object No.	Characteristic Vector
1	[3,2,0,1,2,0,0,0]
2	[4,2,0,4,0,0,0,0]
3	[4,1,0,4,0,0,0,0]
4	[6,2,0,6,0,0,0,0]
5	[6,4,1,3,1,0,1,0]
6	[8,3,1,2,3,0,0,2]

The computation of topology preserving maps between vector spaces can be carried out by using Kohonen neural networks[7]. The self organizing maps get the features data as inputs and cluster them on a 2-dimensional map such that similar ones are placed closer. The topology of the Kohonen's self organizing maps are shown in Fig.7. The network is fully connected and basically, they include a grid of units, typically 2D, each of which receives the input vector and operates on a winner-takes-all basis. For a given input, the network weights are changed according to the rule: find a winner unit, and

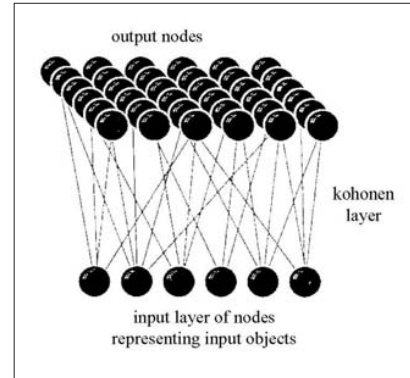


Fig. 7. Kohonen's self organizing maps

change the weights of the units in a neighborhood of the winner unit[9].

Here, self organizing network is trained to recognize the objects by using the following algorithm. During the training process, features are presented several times to train the connection weights. If the object is the object of interest (target), then this will be tracked.

Algorithm 3: [Training]

Step 1. Attribute a random weighting vector to each node. Set up a radius for neighbourhood.

Step 2. Apply the input characteristic vector.

Step 3. Use the following formula to calculate the Euclidean distance d_j between the input vector and each output vector j .

$$d_j = \sum_{i=1}^N (x_i(t) - m_{ij}(t))^2$$
 where N is the number of nodes, $x_i(t)$ is the characteristic vector of the i^{th} object at time t , $m_{ij}(t)$ is the weighting vector of the j^{th} node under the i^{th} input and at time t .

Step 4. Choose the node j which has the minimum d_j .

Step 5. Update the weighting vectors of the node j and its neighbours, in the neighbourhood N_j . Using a gain term $\alpha(t)$ which is selected as $(0 < \alpha < 1)$ and decreases in time

$$m_i(t+1) = m_i(t) + \alpha(t)[x_i(t) - m_i(t)] \forall i \in N_j(t)$$

$$m_i(t+l) = m_i(t) \forall i \notin N_j(t).$$

Step 6. Repeat by going to Step 2 until the termination conditions are satisfied. After training using this scheme, the system has been particularly successful in recognizing the object of interest in cluttered environments.

III. ROBOTIC INTERCEPTION

After identifying the target object and obtaining its initial location by using visual system, the next phase of this study is executing robotic interception process at a convenient rendezvous point.

A. Determining initial position of an object

Fig.8 illustrates the layout of the physical platform in detail. In this system, the camera is located at a position where its lens is perpendicular to the conveyor surface and its field of view covers whole width of the conveyor belt. As depicted in

Fig.8, objects travel on the conveyor until their image touches a virtual detection line in the image plane. As soon as this contact takes place, the frame is captured and processed to extract position of the object from basic geometry:

$$X = \frac{x_i p_x Z}{S_x}$$

$$Y = \frac{y_i p_y Z}{S_y},$$

where p_x, p_y are pixel dimensions, x_i, y_i are image coordinates in the x-y coordinate system, and Z is the distance of the camera to the conveyor surface.

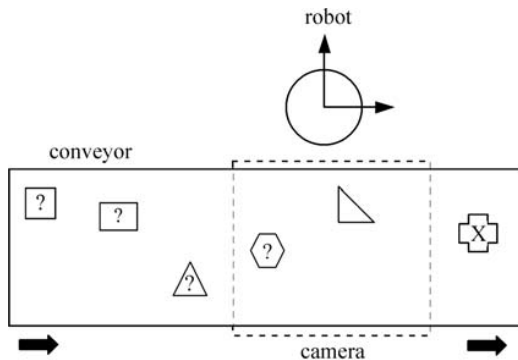


Fig. 8. Layout of the system

B. Rendezvous planning

For intercepting objects traveling along predictable trajectories, Prediction, Planning and Execution (PPE) methods are well suited [10]. In a PPE strategy, the motion of an object through a robot's workspace is predicted. Robot motion to intercept the object is then planned and executed [11]. For the problem at hand, the solution to the rendezvous problem is confined to planar coordinates. The initial position of the target is determined by visual system. The target is moving along with conveyor at a constant speed and robot manipulator is at a location over the conveyor belt, thus movement at third direction can be omitted. Under definite initial conditions of the manipulator and the target, trajectory for robotic interception at a rendezvous point on the conveyor is analytically calculated, as follows.

The initial position of end-effector (x_B, y_B) , visualized target position located as the origin (x_0, y_0) , and planned rendezvous point (R) are shown in Fig.9.

V_c, V_x, V_y and t_R denote speed of the target, end-effector's average speed on x and y directions, and time required for the end-effector to arrive to the rendezvous point, respectively.

$$t_R = \frac{y_B}{V_y} \quad (2)$$

$$x_R = V_x t_R$$

$$x_B + x_R = V_c t_R$$

$$x_R = V_c t_R - x_B$$

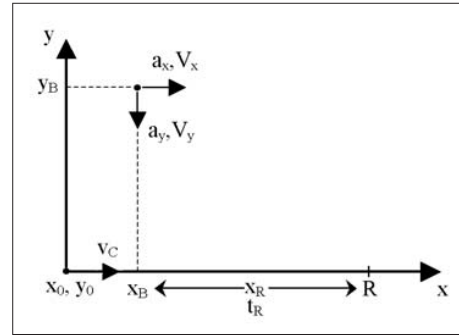


Fig. 9. 2-D reference diagram

$$\frac{V_x}{V_y} y_B = V_c \frac{y_B}{V_y} - x_B$$

$$V_x = V_c - x_B \frac{V_y}{y_B} \quad (3)$$

V_y determines both of the V_x and rendezvous time, thus firstly V_y is determined utilizing the velocity diagram in Fig.10.

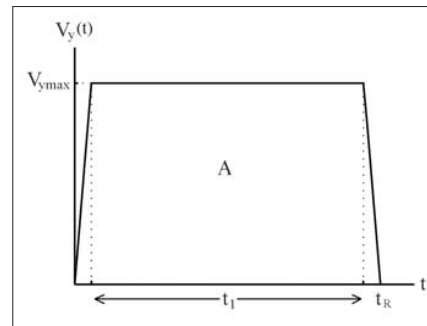


Fig. 10. V_y velocity diagram

Selected acceleration(a) and maximum velocity (V_{ymax}) values determine the variation characteristic of the velocity(V_y). The motion of the end-effector will be determined by using the velocity values. The velocity(V_y) can be formulated as follows:

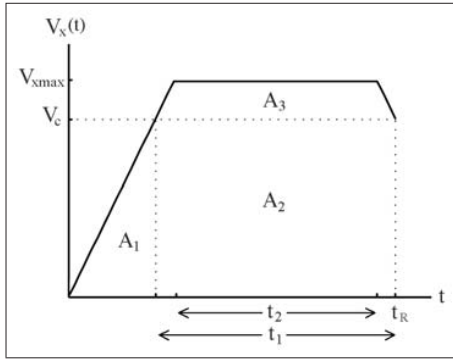
$$A = y_B$$

$$t_1 = \frac{y_B}{V_y} - \frac{2V_{ymax}}{a}$$

$$y_B = \frac{V_{ymax}^2}{a} + \left(\frac{y_B}{V_y} + \frac{2V_{ymax}}{a} \right) V_{ymax}$$

$$V_y = \frac{ay_B V_{ymax}}{ay_B + V_{ymax}^2} \quad (4)$$

Motion of the end-effector on x-direction will be formulated by using Fig.11 as follows:

Fig. 11. V_x velocity diagram

$$x_R = A_1 + A_2 + A_3$$

$$x_R = \frac{1}{2} \frac{V_c^2}{a} + V_c t_R - \frac{V_c^2}{a} +$$

$$\left(t_R - \frac{2V_{xmax}}{a} + \frac{V_c}{a} \right) (V_{xmax} - V_c) +$$

$$\frac{(V_{xmax} - V_c)^2}{a}$$

$$x_R = -\frac{1}{2} \frac{V_c^2}{a} + V_{xmax} \left(\frac{y_B}{V_y} \right) -$$

$$\left(\frac{V_{xmax}}{a} \right) (V_{xmax} - V_c). \quad (5)$$

Utilizing the position on x-direction equation above, V_{xmax} will be expressed with a and V_y variables. Position equation on x-direction can be expressed as below:

$$x_R = V_c \frac{y_B}{V_y} - x_B,$$

using equation above;

$$V_{xmax} = \frac{-k_1 \pm \sqrt{k_1^2 - 4k_2}}{2} \quad (6)$$

where

$$k_1 = -V_c$$

$$k_2 = a \left(V_c \frac{y_B}{V_y} - x_B \right) + \frac{V_c^2}{2} - V_{xmax} \left(\frac{y_B}{V_y} \right).$$

End-effector's equations of motion on each directions at any time t can be found as below:

$$x = \frac{1}{2} at^2 \quad t \leq t_1$$

$$x = \frac{1}{2} \frac{V_{xmax}^2}{a} + (t - t_1) V_{xmax} \quad t_1 < t \leq t_2$$

$$x = \frac{1}{2} \frac{V_{xmax}^2}{a} + V_{xmax} (t_2 - t_1) + \left[V_{xmax} - \frac{a \Delta t}{2} \right] \Delta t$$

$$t_2 < t \leq t_R$$

$$y = \frac{1}{2} at^2 \quad t \leq t_i$$

$$y = \frac{1}{2} \frac{V_{ymax}^2}{a} + (t - t_i) V_{ymax} \quad t_i < t \leq t_{ii}$$

$$y = \frac{V_y V_{ymax}^2}{a(V_{ymax} - V_y)} - \frac{1}{2} a(t_R - t)^2 \quad t_{ii} < t \leq t_R$$

where

$$t_1 = \frac{V_{xmax}}{a} \quad t_i = \frac{V_{ymax}}{a}$$

$$t_2 = t_R - \left(\frac{V_{xmax} - V_c}{a} \right) \quad t_{ii} = t_R - \left(\frac{V_{ymax}}{a} \right)$$

$$\Delta t = t - t_2.$$

IV. EXPERIMENTAL RESULTS

The resulting planned trajectory obtained by using computer simulation can be seen in Fig.12. The application platform consists of a conveyor, a vision system and a robotic manipulator. Object recognition algorithms and the calculations for the planned trajectory are tested by using this platform. Captured images of object tracking and interception by a robotic arm at a preplanned rendezvous point are displayed in Fig.13. Combining the end-effector coordinates from snapshots of the motion show that the end-effector follows the calculated trajectory end up intercepting and grasping the targeted object.

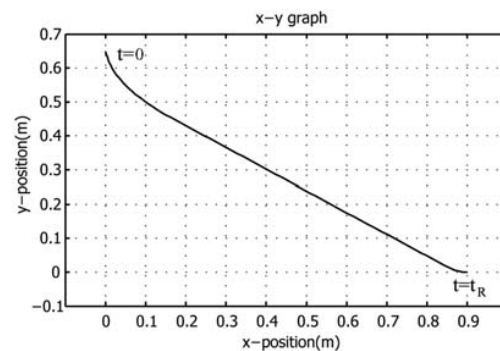


Fig. 12. Simulated end-effector's trajectory

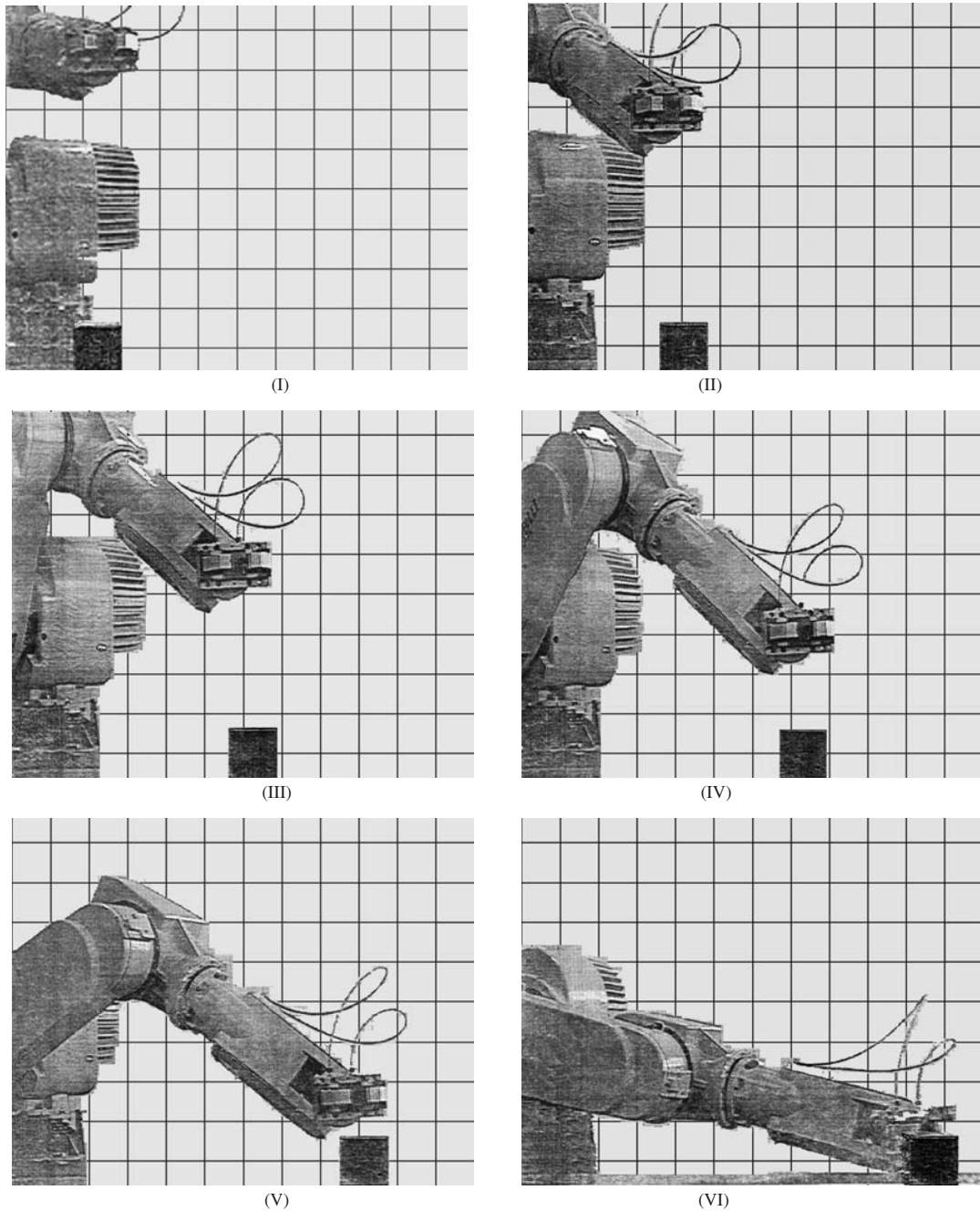


Fig. 13. Experimental results for a robotic interception on a conveyor belt

V. CONCLUSION

In this paper we have addressed an assemblage of visual object recognition and rendezvous planning problems for a robotic manipulation system. The objective was to identify a target on a conveyor belt and plan a convenient trajectory for robotic interception and execute these on a real platform. The key to the success of such a system is the knowledge of the targets location. Since the vision is a useful sensory capability that it allows for non-contact measurement of environment, then visuality of the target adds great effectiveness for the manipulation tasks. The results of the study shows the potential of using robotic vision for industrial purposes.

REFERENCES

- [1] P.I. Corke, *Visual Control of Robot Manipulators—A Review*, in Visual Servoing, edited by K. Hashimoto. World Scientific: Singapore, 1993.
- [2] Y. Shirai and H. Inoue, "Guiding a robot by visual feedback in assembling tasks," *Pattern Recognition*, vol.5, pp.99–108, 1973.
- [3] M. Sitti, M. Ertugrul and A. Denker, "Coordination of two robots using visual feedback," *In Proc. of Int. Conf. on Recent Advances in Mechatronics*, Istanbul, Turkey, 1995, pp.615–620.
- [4] S. Hutchinson, G.D. Hager and P.I. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robot. Autom.*, vol.12, pp.651–670, 1996.
- [5] T. Borangiu, "Visual conveyor tracking for "pick-on-the-fly" robot motion control," *7th International Workshop on Advanced Motion Control*, Maribor, Slovenia, 2002, pp.353–358.
- [6] D. Kragic, M. Bjorkman, H.I. Christensen and J-O. Eklundh, "Vision for Robotic Object Manipulation in Domestic Settings," *Robotics and Autonomous Systems*, vol.52, pp.85–100, 2005.
- [7] T. Kohonen, *Self-organizing Maps*, Third Edition. Springer-Verlag: Berlin, 2001.
- [8] R.C. Gonzales and R.E. Woods, *Digital Image Processing*, Second Edition. Prentice Hall: N.J., 2002.
- [9] G. Coppini, S. Diciotti, and G. Valli, "Matching of medical images by self-organizing neural networks," *Pattern Recognition Letters*, vol.25, pp.341–352, 2004.
- [10] M. Mehrandezh, N.M. Sela, R.G. Fenton and B. Benhabib, "Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique," *IEEE Transactions on Systems, Man and Cybernetics*, vol.30, pp.238–250, 2000.
- [11] D. Hujic, E. Croft, G. Zak, R.G. Fenton, J.K. Mills and B. Benhabib, "The robotic interception of moving objects in industrial settings: strategy development and experiment," *IEEE/ASME Transactions on Mechatronics*, vol.3, pp.225–239, 1998.

Prof.Dr.Ahmet Denker. B.S in EE, Boğaziçi University(1977); M.S. and Ph.D Sussex University, England (1978-1981). Dr. Denker worked as a full time professor at Boğaziçi University between years 1982 and 1996. He was entitled "Chartered Engineer" (C.Eng.) by the Council of Engineering of U.K.. His particular areas of interest are control and information technologies. He held visiting positions in Sussex University (U.K.), Open University (U.K.), Eastern Mediterranean University (Cyprus) and Keio University (Japan). In 1996 he was awarded a medal by Matsumae International Organization of Japan for his contributions to science and peace.

He acted as the General Manager in industry between years 1996 and 2003. Dr. Denker is a full time professor at Ankara University since 2003 and gives lectures on Information Systems Engineering and Management.

Tuğrul Adıgüzel received the B.S. and M.S. degrees from Ankara University, Ankara, Turkey, in 1999 and 2001, respectively, where he is currently toward the Ph.D. degree.

Since 1999, he has been a Research Assistant with the Electronics Engineering Department, Ankara University. His research interests are nonlinear control design and robotics in particular.