Int. J. Engng Ed. Vol. 10, No. 5, pp. 427–433, 1994
Printed in Great Britain.

0742–0269/91 $3.00+0.00

A Low-Cost Robot Vision System to Simulate a Manufacturing Environment*

R. I. NOORANI
Mechanical Engineering Department, Loyola Marymount Univesity, Los Angeles, CA 90045-2699, USA
R. S. STITTS†
Electrical Engineering Department, Loyola Marymount University, Los Angeles, CA 90045-2699, USA

This paper describes a low-cost robot vision system for manipulating objects in a cluttered environment. The vision system identifies objects of different shapes and sizes, sorts them accordingly and puts them onto a conveyor belt to simulate a manufacturing environment. This paper will discuss the theory of vision systems, equipment selected, experimental procedures and conclusions drawn. The robot vision algorithm developed will also be presented.

INTRODUCTION

THE APPLICATION of machine vision as an improvement to robot performance is becoming an essential element for the integration of robotic technology in the manufacturing environment. The machine vision increases the robot's interaction with its work environment and helps obtain greater system flexibility [1]. While robots have been used in manufacturing for over 30 years, only recently have they proliferated into a wide variety of applications.

Most manufacturing applications for robots do not require vision systems. However, there are many applications where a vision system has a distinct advantage and may even be essential for a robot to perform a task. Some typical applications of a robot vision system are (a) arc-welding seams that vary in length, shape and position, (b) spray-painting a series of parts of different sizes and shapes, (c) selecting parts from a mixed group or pile, (d) picking parts from a moving conveyor line and (e) inspecting parts for defects.

This paper describes the development of a low-cost robot vision system that is used to manipulate objects in a cluttered environment. The vision system identifies objects of different shapes and sizes, sorts them accordingly, and places them on a conveyor belt to simulate a manufacturing environment.

Machine vision is a relatively new technology that is undergoing, and will continue to undergo, a process of improvement throughout the next decade [2]. While robotic applications picked up in the 1970s, the development of robotic vision systems began in the 1980s.

* Accepted 2 April 1994.

† Currently, graduate student at Rensselaer Polytechnic Institute.

During recent years, many engineering solutions have been shown to be feasible for manipulation of objects by a robot. For example, if the aim is to pick objects from a bin and the objects' surfaces are smooth, it is possible to use a vacuum gripper [3]. If the aim is somewhat more sophisticated, such as sorting a pile of objects on the basis of shape [4], one must use the sensory data to not only locate the least occluded object but also to recognize its shape in two or three dimensions. We present in this research a hardware and software based control method for manipulating objects.

The paper also emphasizes a personal computer (PC)-based control system, as more and more manufacturing systems utilize PCs for their increased power and flexibility [5].

OBJECTIVES

The main objectives of this research are the following:

(1) To understand the theory, operation and interfacing of a robot vision system.
(2) To develop and use a vision algorithm for the precise manipulation of objects in a cluttered environment.

THEORY

Robot vision may be defined as the process of extracting, characterizing and interpreting information from images of a three-dimensional world. The vision capabilities endow a robot with a sophisticated sensing mechanism that allows it to respond to its environment 'intelligently'.

The overall theory involved in this research can be divided into two topics: *image acquisition* and *image analysis*. Each of these will be discussed in this section.

Image acquisition

Image acquisition is the process of capturing the image of an object using an image device. The image device may be a camera, a photo diode array, charge-coupled device (CCD) array or charge-injection device (CID) array. Whatever the image device may be, the output of the device is a continuous analog signal proportional to the amount of light reflected from the image. The signal is then converted to a digital signal using a digitizer. The digitizer is basically an analog-to-digital converter (ADC) with a circuit to 'freeze' the image.

The image area is then divided into regions called picture elements or pixels (Fig. 1a). The ADC compares the brightness of each pixel and then assigns a value of zero (black) or one (white) to that pixel. This is the data that the computer manipulates to extract information about the image. A one-bit ADC can only distinguish between black and white. An 8-bit ADC can distinguish $2^8 = 256$ (0–255) different shades of gray (Fig. 1b); hence, the term 'gray level'. A high number of pixels results in greater image resolution and better image detail, but it also takes longer for the computer to process the image.

Image analysis

The process of image analysis consists of several operations, the most fundamental of which is the edge detection. In order for the vision system to locate, identify and pick an object, it must first detect the edges of the object. The edges of an object can be used to construct a drawing which ultimately results in image understanding [6]. The process of image understanding is shown in Fig. 2.

The edges of an object are generally represented by the points that exhibit the greatest difference in gray-level values within a picture matrix. This can be achieved by taking the derivative of the gray scale intensity function. Thus, the value of the slope would be high in the location where an edge exists. The Roberts cross operator provides a good

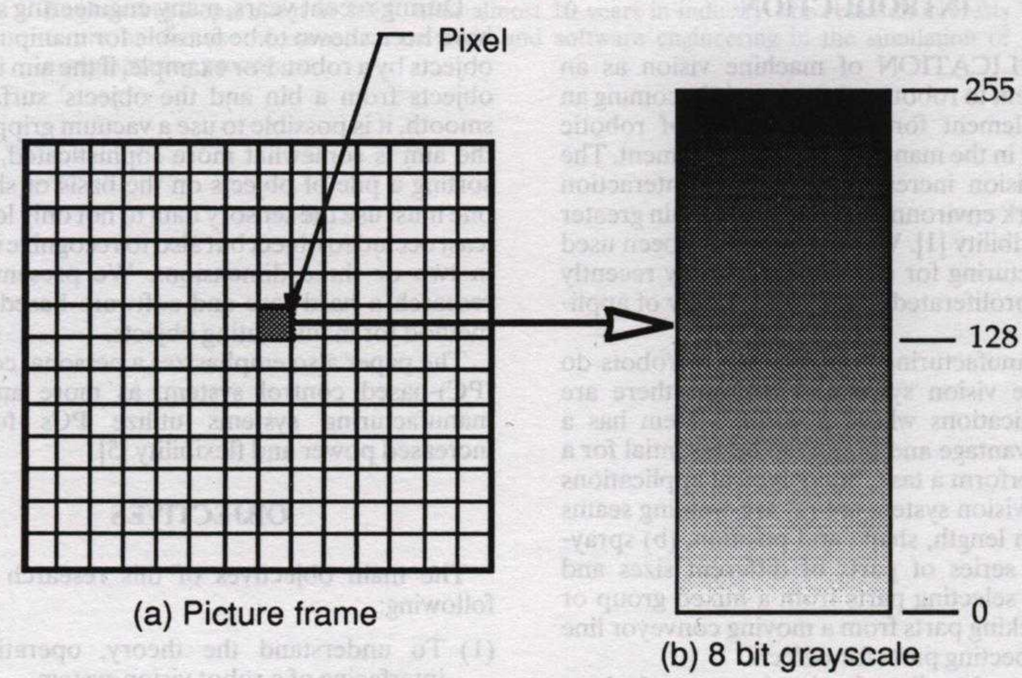

Fig. 1. Example of picture frame and 8-bit gray scale.

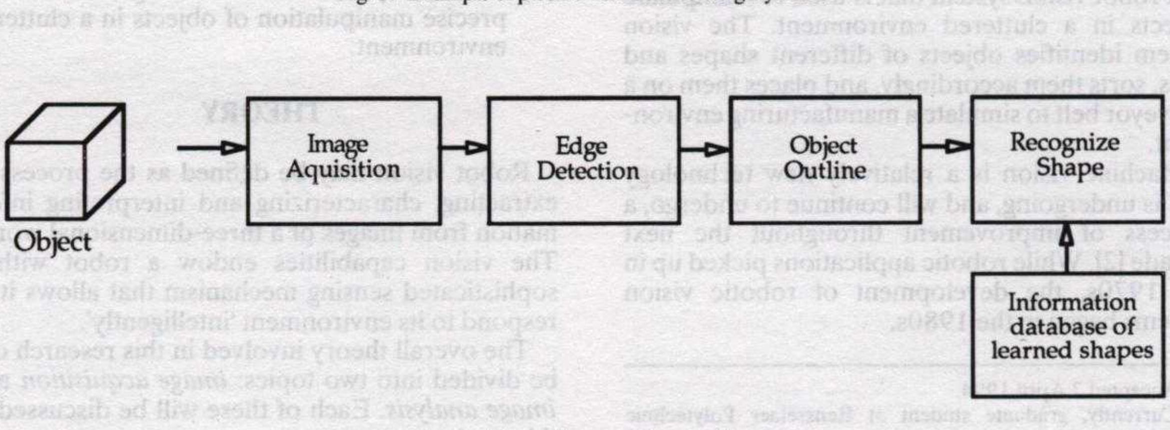

Fig. 2. Steps to image understanding.

approximation of the first derivative of a digital image. The Roberts operator is defined as:

$$R(m, n) = \{[i(m+1, n+1) - i(m, n)]^2 + [i(m, n+1) - i(m+1, n)^2]\}^{1/2} \quad (1)$$

where $i(m, n)$ is the image intensity of the pixel located at (m, n).

Example. The gray-level values of a picture matrix are given below. Construct (a) the gradient matrix using the Roberts cross operator and (b) the binary matrix using a threshold value of three:

$$\begin{bmatrix} 8 & 8 & 8 & 3 \\ 9 & 8 & 4 & 3 \\ 8 & 6 & 5 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

We will solve this using equation (1). Finding the first two terms:

$$\begin{bmatrix} 8 & 8 \\ 9 & 8 \end{bmatrix} \Rightarrow [(8-8)^2 + (8-9)^2]^{1/2} = 1$$

$$\begin{bmatrix} 8 & 8 \\ 8 & 4 \end{bmatrix} \Rightarrow [(4-8)^2 + (8-8)^2]^{1/2} = 4$$

This can be continued to find the rest of the gradient matrix. The gradient matrix for this problem is:

$$\begin{bmatrix} 1 & 4 & 5.1 & x \\ 3 & 3.6 & 2.2 & x \\ 5.8 & 3.6 & 2 & x \\ x & x & x & x \end{bmatrix}$$

Next, we will find the binary matrix using a threshold value of three. An edge of an object is present if the gradient is greater than the threshold. The threshold value is an arbitrary value set based on the needs of the system. Therefore an operator value $\geq 3 = 1$ and an operator value $< 3 = 0$. Below we find the binary matrix for the initial picture matrix. Based on the binary matrix, lines may be identified that represent an object. The lines are drawn between the ones and the zeros on the matrix. These lines outline the shape of the object:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The setup for this experiment is shown in Fig. 3. It consists of (1) a PC for program storage and image analysis, (2) Rhino Mark IV Robot Controller which controls the robot arm, (3) Rhino XR-3 Robot Arm, (4) a camera to see the objects, (5) a frame grabber board (inside PC) which stores the image and translates it into data that the computer can understand, (6) a monitor to display the image from the camera, and (7) a conveyor belt which is used to simulate the manufacturing environment. The robotic arm is a five-axis educational robotic arm which also has a gripper. The software used is RoboTalk from Rhino Robots. This software is necessary for communication between the robot controller, PC and the frame grabber board. RoboTalk is a robotic language similar to those used in industrial robotics.

PROCEDURE

Step 1: setup and calibration

We begin this experiment by connecting the robot controller, robot, vision system hardware and personal computer as per the manufacturers instructions. As the robot is connected, the workcell is set up in an efficient and safe manner, taking into consideration the possible movements of the robot. Next, the robot programming and communications package, RoboTalk, is loaded into the robot. Using this software we can initialize and calibrate the vision system. This must be done each time the system is turned on so that the robot can find its position with respect to the viewing area of the camera. To do this we first use the VISINIT command to initialize the vision board. Next, we use the CALIBRATE command. This command

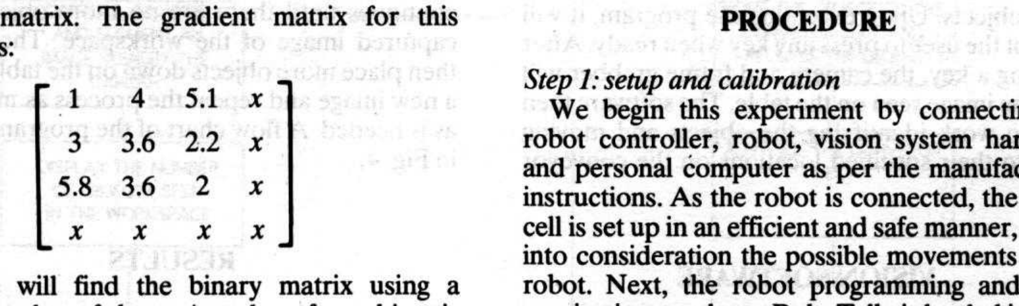

Fig. 3. Experimental setup.

will ask us to place six objects on the screen. The user will then move the robot to those points using the teach pendent. After all six points have been programmed, the robot is ready to use its vision capabilities.

Step 2: teaching the robot what shapes look like

Our next step is to teach the robot what different shapes look like. We do this by using a small program which, when executed, numbers all of the shapes seen by the camera in its viewing area. The user then types in the name of the shape corresponding to the number on the screen. This information is stored in a file on the PC so that this step does not have to be repeated. The computer stores the shapes by their geometric properties such as shape and the location of centroid.

Step 3: running the vision program

To run the vision program that was developed for the rhino robot we now load the main program into the RoboTalk software package. This is the program that will recognize objects and move them to the conveyor belt. The program will reference the datafile created earlier to compare the geometric properties of the objects on the table. The objects do not necessarily have to be the same size as the ones that were originally digitized. The only limitation to size is that the gripper can only hold small objects. Upon executing the program, it will prompt the user to press any key when ready. After pressing a key, the camera and frame grabber will grab the image seen on the table. The software then goes to work identifying the objects and moving them to their specified locations on the conveyor belt.

VISION SOFTWARE

The vision software that has been developed for the robot vision system has two parts. The first part is a program called LEARNSHIP, which memorizes the shapes of objects and stores the shapes for later retrieval. The second program called RPP, is the actual program which utilizes both the vision system and the robot arm for the manipulation of the objects. These programs were written in the RoboTalk language.

There are several important commands contained in RoboTalk that we have used in these programs. The LIVE command tells the vision board to begin taking real-time pictures of the workspace. These pictures are displayed on the screen of the vision system monitor. The CAPTURE command captures an image of the workspace and displays it on the vision monitor. BINARIZE takes the image and turns it into a black and white image so that image boundaries may be found. SEGMENT separates the different objects and creates a list of the objects and their characteristics in the computer. LEARN allows one to assign names to the different objects on the screen. REMEMBER writes the list of information about the objects in the workspace to a file for later retrieval. RECALL is used to retrieve a list of objects from the datafile saved using the REMEMBER command. These are the main commands seen in the program which manipulate the image of the workspace as seen by the camera.

The first program uses the RoboTalk command libraries to make a list of the object names and characteristics and save them to a file called objects which are later retrieved by program 2. The program is listed below.

```
LIVE
CAPTURE
BINARIZE
SEGMENT
LEARN
REMEMBER "objects"
END
```

Program 2 is the main program which captures an image of the workspace and recognizes the objects. It then begins a loop in which it matches each object's characteristics to previously memorized objects. The program then commands the robot arm to pick up the pieces and move them to a conveyor belt on the other side of the robot. Upon completion of the action the loop continues and the robot recognizes and moves to the next object. This continues until there are no more objects in the captured image of the workspace. The user can then place more objects down on the table, capture a new image and repeat the process as many times as is needed. A flow chart of the program is shown in Fig. 4.

RESULTS

The results of this research are the confirmation of the objectives, introducing the readers to the theory, control and application of a robot vision system. The greatest exposure to new materials was in the theory of the vision system and the development of the vision algorithm.

The main program 'RPP' that controls the vision system, detects subtle changes in the light intensity that cannot be seen by the human eye. The listing of the program is shown in Appendix A.

DISCUSSION AND CONCLUSIONS

This project has demonstrated that a low-cost educational robot, when equipped with a vision system, can be used for the integration of robotic technology in the manufacturing environment. The robot vision system endows the robot with a sophisticated sensing mechanism that allows the robot to respond to its environment in an 'intelligent' and 'flexible' manner. While a robot and its vision system can be controlled in many ways,

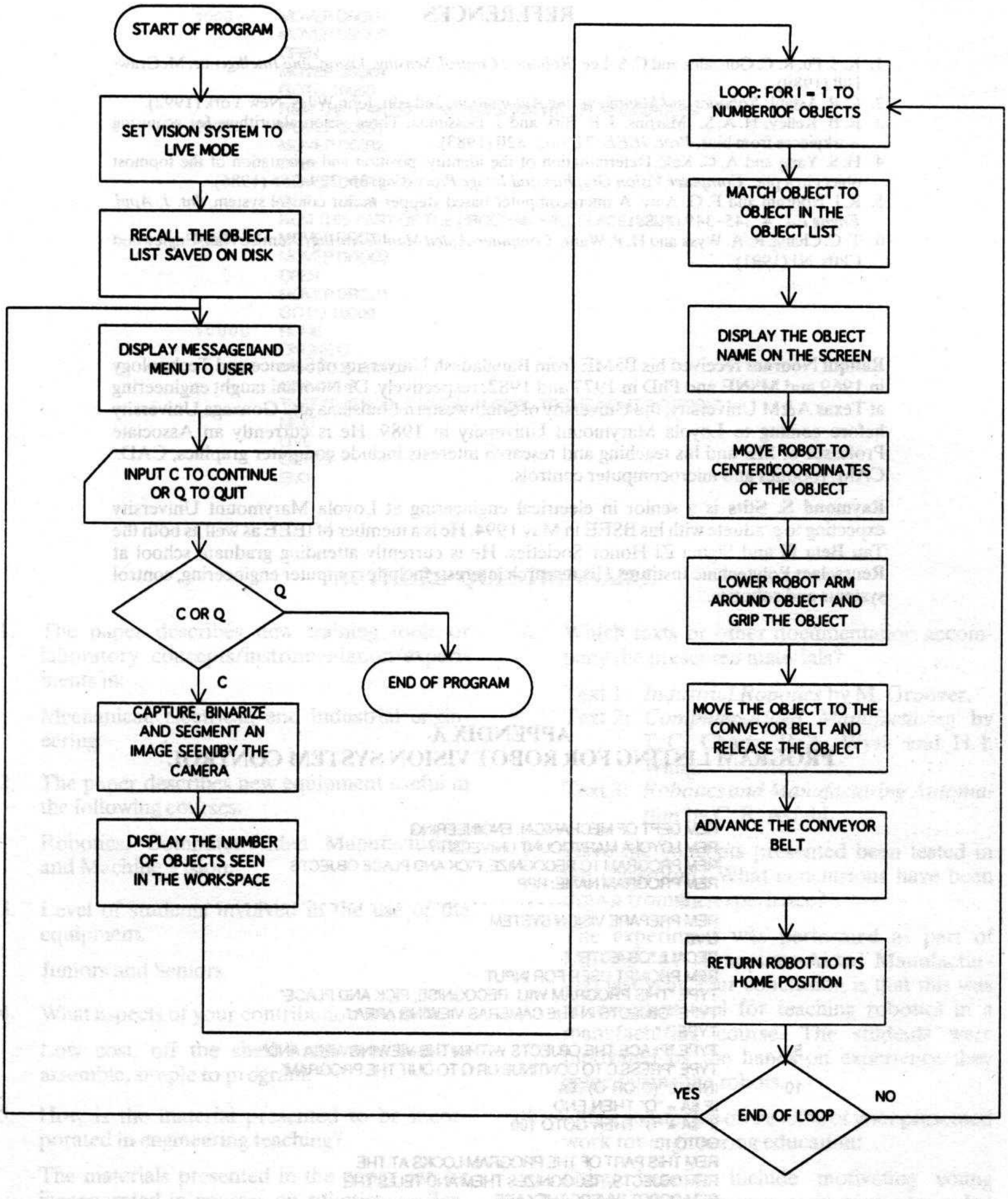

Fig. 4. Flow chart of the main program.

software-based control is a technique that is gaining popularity among reseachers. The key advantage of a programmable robot is flexibility. Such flexibility reduces the cost of operation of a robot system. A programmable robot vision system can be reprogrammed without physically modifying the robot geometry. Flexibility is enhanced by automatically generating the appropriate program.

Acknowledgement—Partial support of this project was provided by the National Science Foundation's Intrumentation and Laboratory Improvement Program through grant no. USE-9050992.

REFERENCES

1. K. S. Fu, R. C. Gonzalez and C. S. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill (1989).
2. C. R. Asfahl, *Robotics and Manufacturing Automation*, 2nd edn, John Wiley, New York (1992).
3. R. B. Kelley, H. A. S. Martins, J. R. Birk and J. Dessimoz, Three vision algorithms for acquiring workpieces from bins, *Proc. IEEE*, **71**, 803–820 (1983).
4. H. S. Yang and A. C. Kak, Determination of the identity, position and orientation of the topmost object in a pile, *Computer Vision Graphics and Image Processing*, **36**, 229–255 (1986).
5. R. I. Noorani and F. G. Amy, A microcomputer based stepper-motor control system, *Int. J. Appl. Engng Ed.*, **4**, 345–349 (1988).
6. T. C. Chang, R. A. Wysk and H. P. Wang, *Computer-Aided Manufacturing*, Prentice Hall, Englewood Cliffs, NJ (1991).

Rafiqul Noorani received his BSME from Bangladesh University of Science and Technology in 1969 and MSNE and PhD in 1977 and 1982, respectively. Dr Noorani taught engineering at Texas A&M University, the University of Southwestern Louisiana and Gonzaga University before coming to Loyola Marymount University in 1989. He is currently an Associate Professor of ME and his teaching and research interests include computer graphics, CAD/CAM, robotics and microcomputer controls.

Raymond S. Stits is a senior in electrical engineering at Loyola Marymount University expecting to graduate with his BSEE in May 1994. He is a member of IEEE as well as both the Tau Beta Pi and Sigma Zi Honor Societies. He is currently attending graduate school at Rensselaer Polytechnic Institute. His research interests include computer engineering, control systems and robotics.

APPENDIX A
PROGRAM LISTING FOR ROBOT VISION SYSTEM CONTROL

```
        REM DEPT OF MECHANICAL ENGINEERING
        REM LOYOLA MARYMOUNT UNIVERSITY
        REM PROGRAM TO RECOGNIZE, PICK AND PLACE OBJECTS
        REM PROGRAM NAME: RPP
        R E M ...................................................
        REM PREPARE VISION SYSTEM
        LIVE
        RECALL "OBJECTS"
        REM PROMPT USER FOR INPUT
        TYPE "THIS PROGRAM WILL RECOGNISE, PICK AND PLACE"
        TYPE "OBJECTS IN THE CAMERAS VIEWING AREA."
        TYPE " "
5       TYPE "PLACE THE OBJECTS WITHIN THE VIEWING AREA AND"
        TYPE "PRESS C TO CONTINUE OR Q TO QUIT THE PROGRAM"
10      INPUT "(C OR Q)",$A
        IF $A = "Q" THEN END
        IF $A = "C" THEN GOTO 100
        GOTO 10
        REM THIS PART OF THE PROGRAM LOOKS AT THE
        REM OBJECTS, RECOGNIZES THEM AND TELLS THE
        REM ROBOT WHERE THEY ARE
100     CAPTURE
        BINARIZE
        SEGMENT
        TYPE OBJECTS
        FOR I = 1 TO OBJECTS
        MATCH I
        TYPE "THE CURRENT SHAPE IS:"
        TYPE I.NAME
        MOVEX TO I.CENTERX,I.CENTERY,200,0,I.ANGLE
        MOVEX  0,0,-100,0,0
        MOVEX  0,0,-95,0,0
        GRIP
        TYPE "THE OBJECT WILL NOW BE MOVED TO IT'S CORRECT POSITION"
        IF I.NAME = "SQUARE" THEN GOTO 1000
        IF I.NAME = "CIRCLE" THEN GOTO 2000
        IF I.NAME = "ROD" THEN GOTO 3000
        REM THIS PART OF THE PROGRAM WILL PLACE SQUARES
```

```
1000    MOVEP DSQU1
        MOVEP DSQU2
        OPEN
        MOVEP DSQU1
        GOTO 10000
        REM THIS PART OF THE PROGRAM WILL PLACE CIRCLES
2000    MOVEP DCIR1
        MOVEP DCIR2
        OPEN
        MOVEP DCIR1
        GOTO 10000
        REM THIS PART OF THE PROGRAM WILL PLACE RODS
3000    MOVEP DROD1
        MOVEP DROD2
        OPEN
        MOVEP DROD1
        GOTO 10000
10000   HOME
        ONLINE G
        MOVEGH 2500,0
        OFFLINE G
        TYPE "THE ROBOT HAS RETURNED TO THE HOME POSITION"
        NEXT
        LIVE
        GOTO5
        END
```

AUTHORS' QUESTIONNAIRE

1. **The paper describes new training tools or laboratory concepts/instrumentation/experiments in:**

   Mechanical, electrical, and industrial engineering.

2. **The paper describes new equipment useful in the following courses:**

   Robotics, Computer-Aided Manufacturing, and Machine Vision.

3. **Level of students involved in the use of the equipment.**

   Juniors and Seniors.

4. **What aspects of your contribution are new:**

   Low cost, off the shelf hardware, easy to assemble, simple to program.

5. **How is the material presented to be incorporated in engineering teaching?**

   The materials presented in the paper can be incorporated in courses on robotics, or flexible manufacturing. The materials can be used as part of an experiment on robot vision systems.

6. **Which texts or other documentation accompany the presented materials?**

   Text 1: *Industrial Robotics* by M. Groover.
   Text 2: *Computer-Aided Manufacturing* by T. C. Chang, R. A. Wysk and H. I. Wang.
   Text 3: *Robotics and Manufacturing Automation* by C. R. Asfahl.

7. **Have the concepts presented been tested in the classroom? What conclusions have been drawn from the experience?**

   The experiment was performed as part of MECH 451 (Computer-Aided Manufacturing) last year. Our conclusion is that this was an excellent tool for teaching robotics in a manufacturing course. The students were motivated by the hands-on experience they gained using the robots.

8. **Other comments on benefits of your presented work for engineering education:**

   Other benefits include motivating young people to pursue careers in computer-aided manufacturing. The information presented will be useful to those interested in developing similar robot vision experiments.