Journal of Theoretical and Applied Mechanics, Sofia, 2015, vol. 45, No. 1, pp. 53–68

DE GRUYTER
OPEN

DOI: 10.1515/jtam-2015-0004

BUILDING AND PROGRAMMING A SMART ROBOTIC SYSTEM FOR DISTINGUISHING OBJECTS BASED ON THEIR SHAPE AND COLOUR[*]

T. M. Sharari

Institute of Engg & Tech, Dep. of Ele.eeng, Control Lab.& Aut.,
Po. Box. 81. Jordan
e-mail: tsharari@yahoo.com

[Received 26 May 2014. Accepted 17 November 2014]

Abstract. This paper presents a robotic system designed for holding and placing objects based on their colour and shape. The presented robot is given a complete set of instructions of positions and orientation angles for each manipulation motion. The main feature in this paper is that the developed robot used a combination of vision and motion systems for holding and placing the work-objects, mounted on the flat work-plane, based on their shapes and colorus . This combination improves the flexibility of manipulation which may help eliminate the use of some expensive manipulation tasks in a variety of industrial applications. The robotic system presented in this paper is designed as an educational robot that possesses the ability for holding-and-placing operations with limited load. To process the various instructions for holding and placing the work objects, a main control unit - Manipulation Control Unit (MCU) is used as well as a slave unit that performed the actual instructions from the MCU.

Key words: Vision system, colour recognition, positions and orientations, robot manipulator.

1. Introduction and related work

Robotic systems are widely used for repeated tasks whereby their operations are handled by simple position control techniques. To develop the applications of hold-and-place robots, it is important to automate its manipulation process [1, 2, 3, 4, 5, 6, 7]. Also, vision is considered to be the most valuable sense of automating a robotic system. Integration of a vision system

[*] I would like to thank Engs. O. Shoubaki and S. Almasri for their help in carrying out this work.

in the workspace of the robot provides a higher flexibility for adapting to different task requirements. It makes robotic system more versatile by allowing it to deal with variations in work-objects position and orientation [8]. A vision system analyses and produces descriptions of what is imaged. In general, it takes in raw data and generates description or understanding as an output. The technique in robot vision is a combination of sensing and perception [9, 10]. Sensing is the process of acquiring the image into the computer memory and vision perception refers to the observation, collection, processing, and understanding of information from spatial measurements.

This work considers the approach taken to study flexible manufacturing concepts using a combination of vision and motion. It is carried out as part of development of a smart flexible manufacturing system (SFMS). The SFMS is to include an automatic guided vehicle (AGV) which will be transferring work-objects, to be either milled or drilled, to the drilling and milling workstations. A hold-and-place robot is employed to hold and place work-objects to/from the AGV from/to the workstations. It involves manipulation of work-objects with shapes ranging from rectangle, circle, and triangle and with possible colours of red, green, blue, yellow, and black, using a five degree-of-freedom hold-and-place robot. Information obtained from an overhead mounted charged coupled device (CCD) camera is processed to differentiate the work-objects with respect to its shape and colour. The extracted position and orientation information of the work-piece is then utilized to plan the motion of the robot such that the work-piece can be manipulated using its two- fingered gripper. The work-objects to be manipulated are placed at arbitrary positions and orientations on the two- dimensional work-plane of the robot covered by the field of view of the camera. On a user selection, using a graphical user interface (GUI), of the shape and colour of the work-object to be manipulated, the robot, which has been appropriately programmed, then determines the angular rotation of its arms to hold the work-piece from the work-plane and place it at a predefined position. Figure 1 shows the architecture of the overall system. The vision system is equipped with an overhead CCD camera and a structured lighting system. It is linked to the MCU, which handles the vision processing and motion planning of the robot. A GUI integrates the vision system with the hold-and-place robot and also provides a user interface to the integrated system. The enhanced parallel port (EPP), through an interface card, is used as the communication medium between the MCU and the slave unit. A PIC microcontroller acts as a slave unit that carries out required motions of the robot based on commands from the MCU. The robotic system is basically a five degree-of-freedom arm excluding the gripping/releasing movements. The joints of the arm are actuated using stepper motors and DC motors. Sensors in the form of a force sensor and a limit switch handle the gripping and releasing of the work- objects, respectively. The paper is divided into the following sections: 2. vision system; 3. hold-and-place robot; 4. experimentation and results and 5. conclusions.

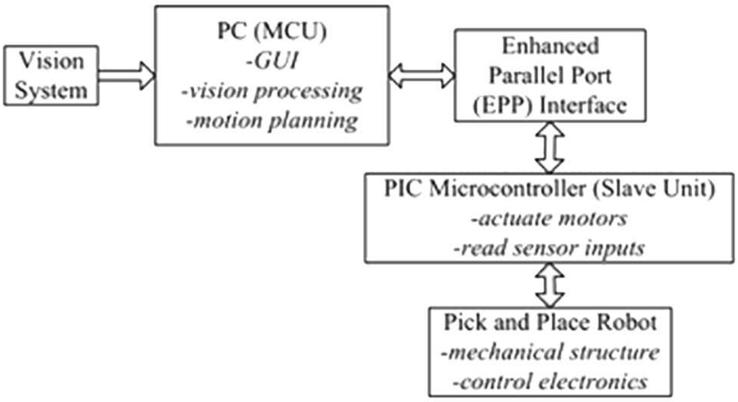


Fig. 1. Architecture of the vision-based hold-and-place robot

2. Vision System

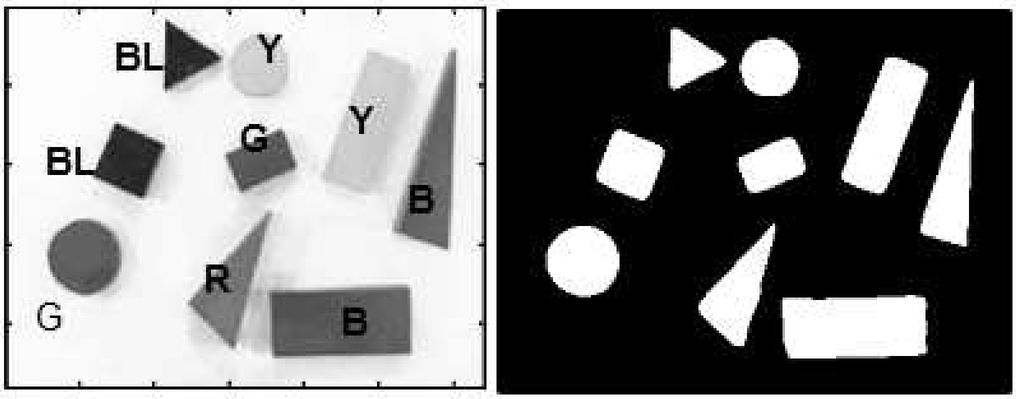
2.1. Image Acquisition

The vision system includes a Sony digital video camera (model DCR HC42E) connected via the USB port to a PC running under the Windows XP platform. It is based under a structured lighting system such that specular reflections from the work-plane and the work-piece surface and shadows around the work- objects could be minimized. Image acquisition is performed in MATLAB using the image acquisition toolbox [11]. The rectangular view, as seen by the camera is streamed into the MCU and a single snapshot image is taken for further processing. The captured image is represented using 320×240 pixels and is in the RGB colour format. That is, a single pixel is represented using the primary red, green, and blue colour values with 8-bit of information.

2.2. Image Processing

Feature extraction for shape recognition requires that the captured image be represented in binary format. That is, with pixel values of either 0 or 1 indicating either a black or a white pixel respectively. Conversion from the RGB colour format to the binary format follows spatial filtering, greyscale conversion, histogram equalization, and finally binary conversion. Spatial filtering performs neighbourhood operations on image pixels, where the value of a filtered pixel depends on the value of its neighbouring pixels. This is achieved through correlation where the centre of the correlation filter, determined through trials with different size and coefficients, is superimposed on the pixel to be filtered. The coefficient of the correlation filter is then multiplied with the pixel value. It is imposed on and the summation of the individual products estimates the value of the filtered pixel.

The filtered RGB image is then converted to a greyscale image, also known as an intensity image. It is in this case represented using 8-bit information where 0 represents black and 255 represents white. A greyscale image often has most of its intensity values concentrated within a particular range. Histogram equalization is performed to enhance the contrast of the greyscale image by transforming the intensity values of the greyscale image so that the histogram of the output image has the intensity values evenly distributed. This enhanced image is finally converted to a binary image using a threshold where values greater than and equal to the threshold are converted to a pixel value of 1 (white) and values below the threshold are converted to a pixel value of 0 (black). A sample captured image with its corresponding binary image is shown in Figs 2 a and 2 b, respectively.



(a) (b)

Fig. 2. (a) RGB format colour image (R for red, G for green and B for blue, BL for black and Y for yellow) and (b) its corresponding binary image

The binary image is then segmented into different regions representing the number of objects, termed as image segmentation. This is done by determining the number of white pixels that are connected to each other which in turn denotes a connected component or object. The image processing steps are implemented using the image processing toolbox in MATLAB [12].

2.3. Feature extraction

2.3.1. Shape recognition

The feature of corner detection is utilized for shape recognition, based on the range of shapes to be differentiated. This is a very realistic approach to the problem of shape recognition, because it is usually the number of sides or corners that humans use for differentiating shapes. A corner, as defined by [13], is a location on the boundary of an object where the curvature becomes unbounded and is defined as:

$$|k(t)|^2 = \left(\frac{d^2y}{dt^2}\right)^2 + \left(\frac{d^2x}{dt^2}\right)^2, \tag{1}$$

where: t represents the distance along the boundary of the segmented regions and a corner is declared whenever $|k(t)|$ assumes a large value. Equation (1) can be modified for a digital binary image. If

$$\Delta x_z = x_{z+s} - x_z \tag{2}$$

$$\Delta y_z = y_{z+s} - y_z \tag{3}$$

where s is the sample length of the curvature in pixels, and with the approximation that

$$\Delta t_z = (\Delta x_z^2 + \Delta y_z^2)^{1/2} \tag{4}$$

then, the curvature of a digital binary image with boundary pixels $z = 1, 2, 3, \ldots,$ Z, for a region, is given as:

$$|k(t)|^2 = \left(\frac{\Delta y_{z+1}}{\Delta t_{z+1}} - \frac{\Delta y_z}{\Delta t_z}\right)^2 + \left(\frac{\Delta x_{z+1}}{\Delta t_{z+1}} - \frac{\Delta x_z}{\Delta t_z}\right)^2 \tag{5}$$

A sample length of five pixels is chosen, through trials with different sample lengths. For best results, the curvature for a sample rectangle, circle, and triangle sampled at five pixels is shown in Figs 3(a), 3(b) and 3(c), respectively. The four and three outstanding peaks in Figs 3(a) and 3(c) correspond to the corners of a rectangle and a triangle, respectively. For a circle, Fig. 3(b), the curvature is nearly constant implying the non-presence of corners.

2.3.2. Colour recognition

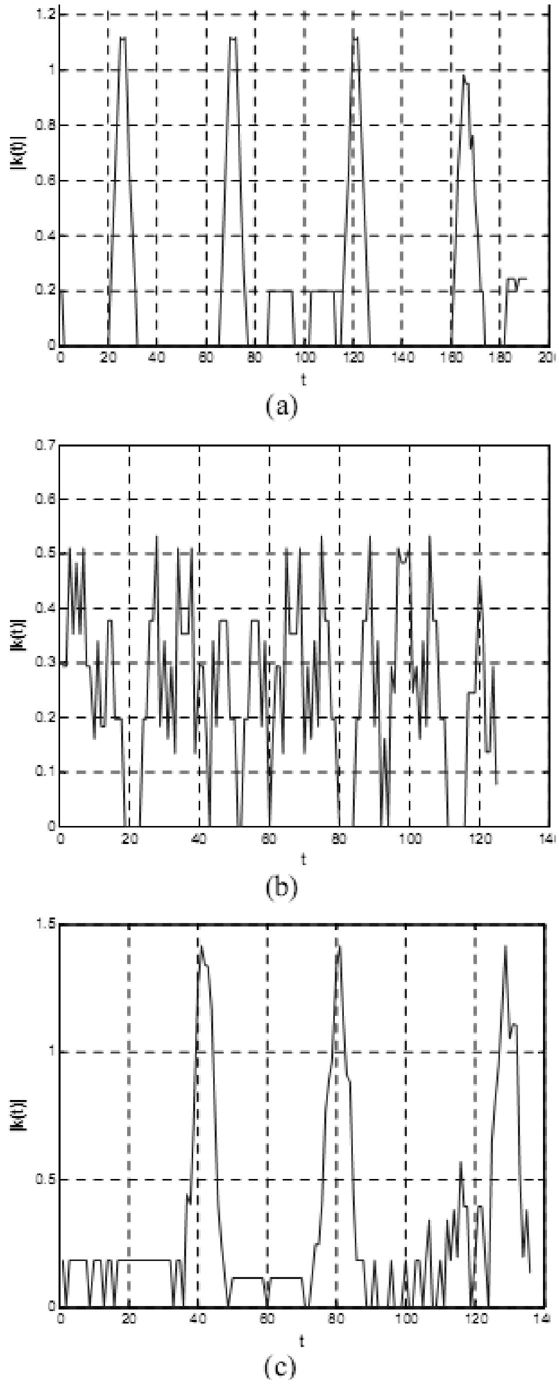
Various colour spaces are used for representing colour images for image processing. Colour spaces are where colours are specified by points in three-dimensional spaces. The colour spaces HSV (hue, saturation, and value) and HSI (hue, saturation, and intensity) are a more natural way to how humans perceive colour and is more often used for colour image processing, than any other colour space. Due to its compatibility with the image processing toolbox, the HSV colour space, shown in Fig. 4, has been preferred. The HSV colour model has colours defined inside a hex cone as shown in Fig. 4 a with the positioning of the hue and saturation in the model, for an arbitrary point, shown in Fig. 4 b. *Hue* describes the colour type given as an angle from 0° to 360°. Typically 0° is red, 60° is yellow, 120° is green, 180° is cyan, 240° is blue, and 300° is magenta. *Saturation*, the purity of a colour or the amount of white added to the colour, has values from 0 to 1, where $S = 1$ specifies a pure colour, that is, no white. *Value* is referred to as the brightness of a colour and ranges from 0 to 1, where 0 is black. The transformation from the RGB colour space to the HSV colour space is given in [14].

2.4. Feature Classification

Feature classification aims to categorize an object within an image to a particular feature class from a number of possible feature classes. A feature class is defined as a group of objects that share common feature or properties. The classification system is operated in two modes: training (learning) and classification (testing). In the training mode, a database of images is built, that contains known work-objects. The knowledge gained from the extracted features forms the database for known work-objects with which the classifier is trained to distinguish the different feature classes in the feature space.

2.4.1. Training

Since the number of shapes and colours to be distinguished is three and five, respectively, there are fifteen different possible feature classes. For shape recognition, the priori knowledge is utilized, that the number of corners for a rectangle, triangle, and circle are always four, three, and zero, respectively. However, for colour recognition, the H, S, and V colour values are obtained through samples of forty- five work-objects to represent a feature class. The



(a)

(b)

(c)

Fig. 3. Curvature for sample of (a) rectangle, (b) circle and (c) triangle

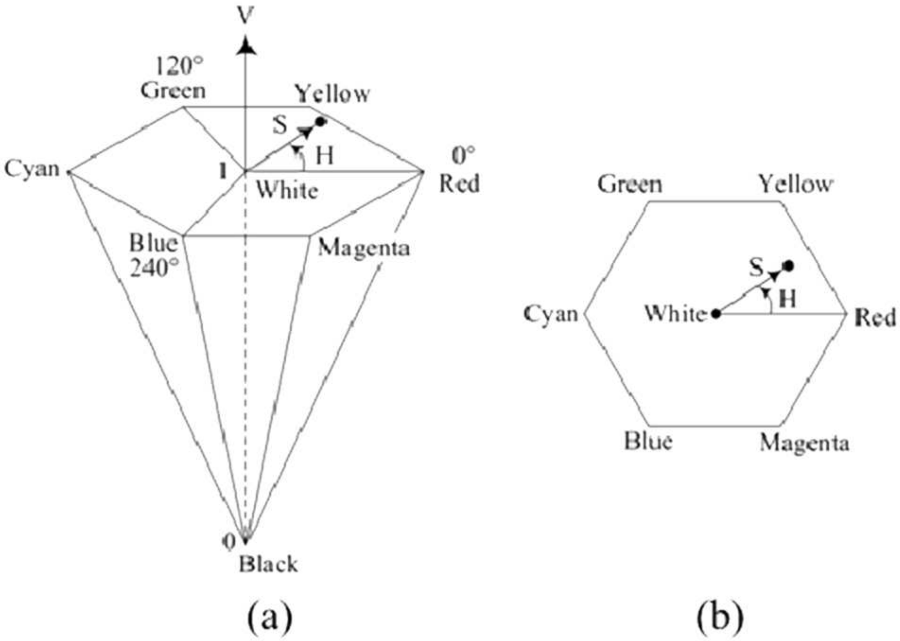


Fig. 4. (a) The HSV colour model; (b) hue and saturation in the HSV colour model

mean RGB and HSV colour values are given in Table 1. Also, the ideal RGB values for the five colours are presented.

Table 1. The mean RGB, ideal RGB colour values and HSV colour values

Colour	Mean RGB colour values	Ideal RGB colour values	Mean HSV colour values
Red	[173.8, 64.0, 47.6]	[255, 0, 0]	[0.0271, 0.7260, 0.6816]
Green	[44.2, 88.1, 57.9]	[0, 255, 0]	[0.3850, 0.4984, 0.3454]
Blue	[47.7, 75.2, 142.9]	[0, 0, 255]	[0.6184, 0.6661, 0.5603]
Yellow	[176.5, 192.8, 19.7]	[255, 255, 0]	[0.1824, 0.8967, 0.7561]
Black	[45.5, 49.3, 46.3]	[0, 0, 0]	[0.5354, 0.2304, 0.2245]

2.4.2 Testing

In the testing phase, an input feature vector is assigned to one of the fifteen feature classes. The simplest approach of minimum distance classification is utilized for feature classification due to the distinct nature of the feature classes. A minimum distance classifier assigns an unknown feature vector, based on the minimum distance between the unknown feature vector and each of the feature classes. The distance here refers to the Euclidean distance, given as:

$$d_j(x) = \|x - m_j\|\,, \quad j = 1, 2, \ldots, W, \tag{6}$$

where $\|a\| = a^T a^{0.5}$ is the Euclidean norm, x denotes the unknown feature vector and m_j denotes the jth feature class [15]. Hence, a feature class x is assigned to class w_i if $d_j(x)$ is the smallest distance.

3. Hold-and-Place robot

3.1. Mechanical design

The hold-and-place robot has been designed and built in-lab. It is designed for a payload of 500 g and is constructed of aluminium due to its light weight. The joints of the robot include the base, shoulder, elbow, wrist, and gripper (rotates about the wrist). The gripper is used as an end-of-arm-tool (EOAT), which grips and releases work-objects. The operation of the gripper is based on the principle of lead screw. One finger of the gripper is fixed while the second finger moves along the lead screw as it is rotated. The calculated values of the torque for all the joints are given in Table 2.

Table 2. Torque considerations for each movement

Joint	Length/radius of joint link (mm)	Torque (mm)
Base	$R_B = 75$	7.79
Shoulder	$L_S = 181$	2.79
Elbow	$L_E = 185$	1.87
Wrist	$L_W = 150$	0.66
Gripper (rotate)	$R_G = 25$	0.66
Grip/release	—	0.013

Stepper motors, due to their good position integrity, are utilized for rotating the base, shoulder, elbow, and the wrist. All the stepper motors have been mounted on the base, so that the torque on the joints is reduced. Transformation of the torque from the motors to the joints is performed, using timing belts and timing pulleys. However, the DC motors to rotate the gripper about the wrist and to open/close the gripper have been mounted on the gripper itself. The gripper is also equipped with a force sensor and a limit switch, that govern the gripping and releasing of a work-piece, respectively. The robot operates on a work-plane with dimensions of 348.14 mm and 261.11 mm. This is determined through the maximum reach of the robot and the ratio of the x and y pixels (320:240 $=$ 4:3) of the captured image.

3.2. Electronics design

An EPP interface card is utilized for communication between the MCU and the slave unit for this system. The EPP interface card offers two 8-bit ports of input and output. A PIC 16F877 microcontroller is used as a slave unit, which operates based on instructions from the MCU. These instructions are in the form of the joint to be actuated, its direction of rotation, and the angle it has to be rotated. The four-phase, 1.8°, hybrid stepper motors are driven using unipolar drive boards, commercially available driver boards for stepper motors. In addition, the light duty DC motors are controlled, using the quadruple half-H driver (SN754410) IC. Furthermore, the force sensor and the limit switch are mounted on the gripper. The output of the force sensor is resistance, whereby applying a force to the active sensing area causes the resistance at the terminals of the force sensor to decrease. Using voltage division, this output resistance is then converted to a voltage between 0 V and 5 V which, based on calibration, depicts a force between the work-object and gripping fingers.

3.3. Motion planning

Motion planning of the robot is performed, based on the predefined home (initial) and place positions, while the hold position is determined after analysis of the captured image. The centroid of the work-object in pixels (x_p, y_p) is first transformed into the dimensions of the work-plane $(x_w,\, y_w)$ and then, to the C-space of the hold-and-place robot (x_r, y_r), as shown in Fig. 5. The z component is always constant as the gripper points vertically downwards, when holding or placing work-objects.

The angular rotation of the base is then defined as:

$$\theta_B = \tan^{-1}\left(\frac{x_r}{y_r}\right). \tag{7}$$

Inverse kinematics is utilized to determine the angular positions of the shoulder, elbow, and wrist at the hold position as shown in Fig. 6

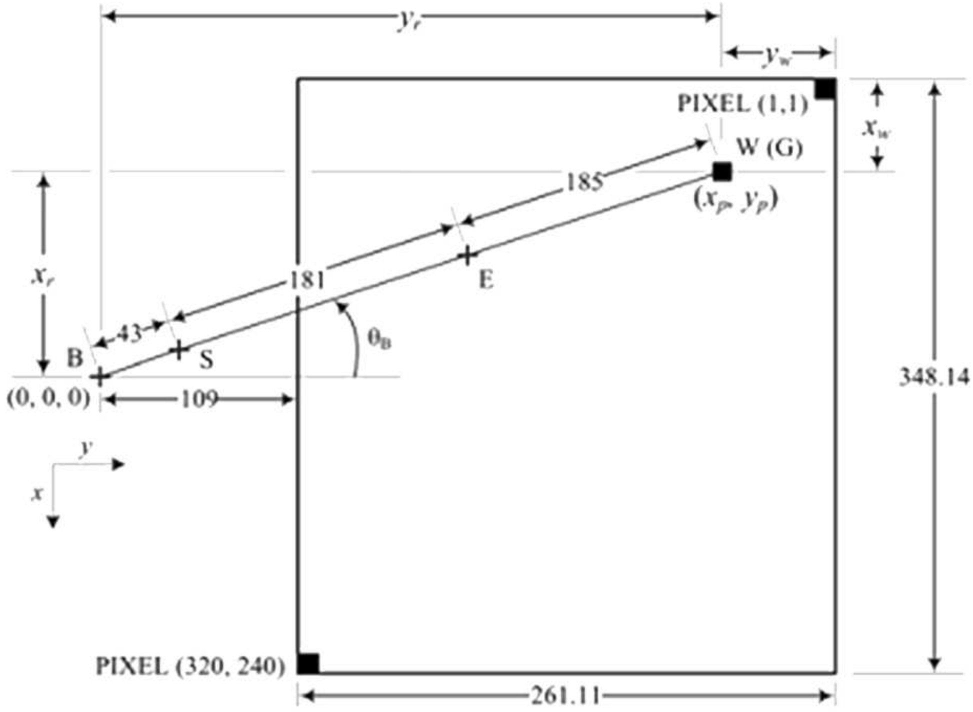


Fig. 5. Representation of the hold position

The angle that the elbow link makes with the shoulder link, θ, is then given as

$$\theta_E = 180^\circ - \beta \tag{8}$$

where

$$\beta = \cos^{-1}\left(\frac{H^2 - (L_S^2 + L_E^2)}{-2L_S L_E}\right) \tag{9}$$

and with reference to Figure 5

$$H = (x_r^2 + y_r^2)^{0.5} - 43. \tag{10}$$

where 43 is the distance between the base joint and the shoulder joint. Using the sine formula, the angle that the shoulder link makes with the xy-plane is

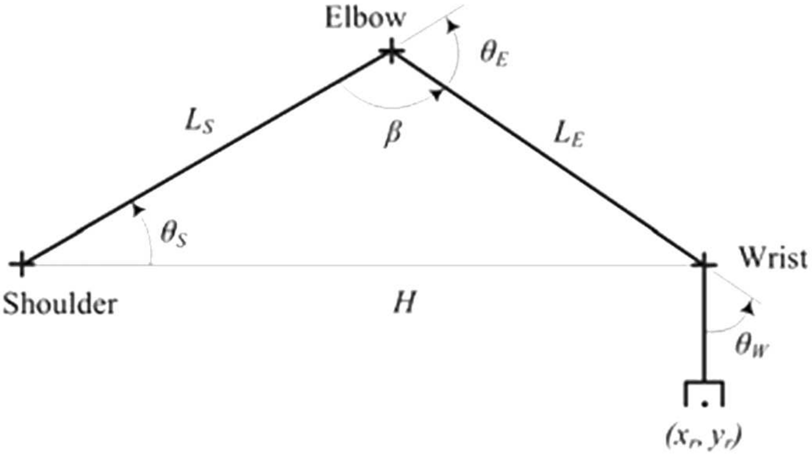


Fig. 6. Angular position of the shoulder, elbow and wrist at the hold position

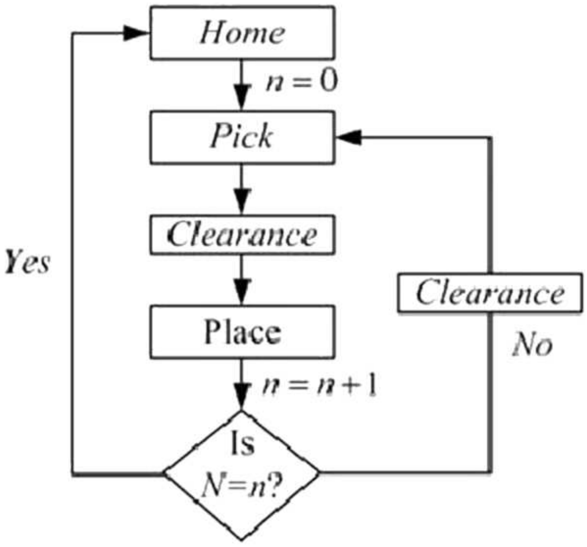


Fig. 7. Motion planning of the robot

given as:

$$\begin{gathered}\frac{\sin\theta_S}{L_E} = \frac{\sin\beta}{H}\\ \theta_S = \sin^{-1}\left(\frac{L_E\sin\beta}{H}\right).\end{gathered}\tag{11}$$

The angle that the wrist link makes with the elbow link is now defined as

$$\theta_W = 90^\circ - (\theta_E - \theta_S). \tag{12}$$

The flowchart for motion planning is illustrated in Fig. 7, where n and N denote the number of work- objects manipulated and the number of work-objects in the specified class. From the home position, the robot holds the first work-piece, and moves through a clearance position to the place position. A clearance position is utilized since the hold and the place heights are the same and it is not feasible to directly rotate the base, with the arm, to/from the hold and place positions. The hold-to- place and place-to-hold motions will continue until all the work-objects of the selected class are manipulated, and then the robot returns to the home position. Figures 8(a), 8(b) and 8(c) show the position of robot at home, hold and place positions, respectively.

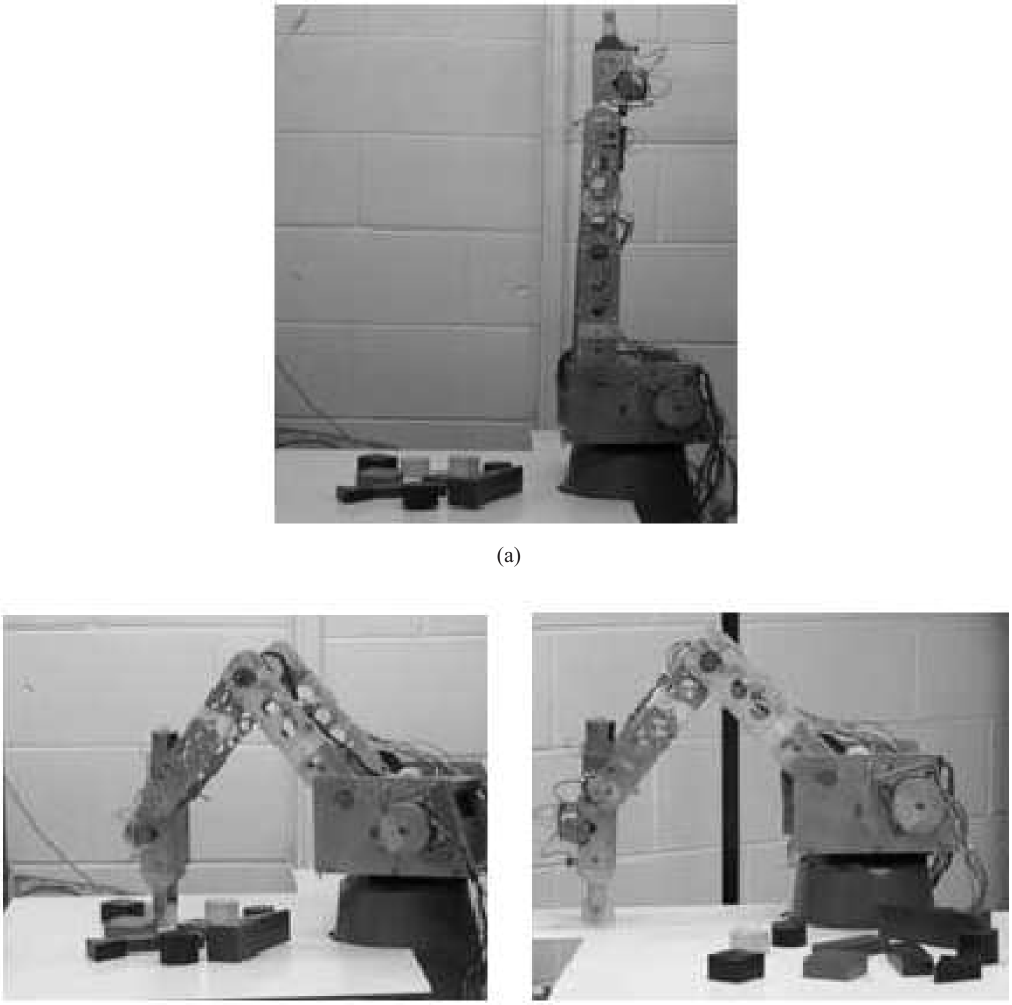
The robot operates in a calibrated environment. That is, the camera has been focused to the field of view of the rectangular work-plane. In addition, the joints, actuated using the stepper and DC motors, work based on calibration, where a certain number of pulses imply a certain degree of movement of the arm link. However, calibration is not used for the gripper motor, as it is controlled by the sensors mounted on it.

4. Experimentation and results

The vision system gives 100% accuracy for both shape and colour recognition for the range of work- objects, considered in this work. For the hold-and-place robot, the accuracy of the joints is given in Table 3. The elbow joint has two operation regions since it had different calibration relationship in its overall operation region.

Table 3. Results for the accuracy of the joints

Movement/rotation		Error (°)	Repeatability ($\pm 3\sigma$)
Base		$e_B = 1.18$	±0.27
Shoulder		$e_S = 0.59$	±0.30
Elbow	Region 1	$e_{E_1} = 2.45$	±0.33
	Region 2	$e_{E_2} = 1.58$	±0.27
Wrist		$e_W = 1.29$	±0.42



(a)

(b)

(c)

Fig. 8. The robot at (a) home, (b) hold and (c) place positions

5. Conclusion

This paper presents a robotic system designed for holding and placing objects based on their colour and shape. The presented robot is given a complete set of instructions of positions and orientation angles for each motion of manipulation. The main feature in this paper is that the developed robot used a combination of vision and motion systems for holding and placing the work-objects, mounted on the a flat work-plane, based on their shapes and colours. This combination improves the flexibility of manipulation, which may help eliminate the use of some expensive manipulation tasks in a variety of industrial applications. The work carried out has been very effective in studying flexible manipulation approaches, using a combination of vision and motion.

REFERENCES

[1] Aliasgar, O. K. Sastry, S. B. Hampapur, S. S. Kamat, V. K. Vyshak. Automated Pick and Place System, Proceeding of 2010 International Conference on Mechanical and Electrical Technology (ICMET), Singapore, September 10–12, 2010, 682–686.

[2] Ali, H., H. H. Low, C. S. Tei. Design and Development of Smart Gripper with Vision Sensor for Industrial Application, Proceeding of 3rd International Conference on Computational Intelligence, Modelling & Simulation, September 20–22, 2011, Langkawi, Malaysia, 2011, 175–180.

[3] Ramaiah, P. S., M. Venkateswara Rao, G. V. Satyanarayana. A Microcontroller based Four Fingered Robotic Hand. *International Journal of Artificial Intelligence & Applications (IJAIA),* **2** (2011), No. **2**, 90–102.

[4] Yussof, J. Wada, M. Ohka. Analysis of Tactile Slippage Control Algorithm for Robotic Hand performing Grasp-move-twist Motions. *International Journal on Smart Sensing and Intelligent Systems,* **3** (2010), No. **3**, 359–375.

[5] Qian, K., X. Ma, X. Z. Dai, F. Fang. Spatial-temporal Collaborative Sequential Monte Carlo for Mobile Robot Localization in Distributed Intelligent Environments. *International Journal on Smart Sensing and Intelligent Systems,* **5** (2012), No. **2**, 295–314.

[6] Nagchaudhuri, A., S. Kuruganty, A. Shakur. Mechatronics Education using an Industrial SCARA Robot, Proceedings 7th. Mechatronics Forum International Conference and Mechatronics Education Workshop (Mechatronics 2000), GA, Atlanta, 2000, 305–311.

[7] Perez, T. L., J. L. Jones, E. Mazer, P. A. O'Donnell. Task-level Planning of Hold-and- place Robot Motions. *IEEE Computer*, **22** (1989), No. **3**, 21–29.

[8] Spong, M. W. Motion Control of Robot Systems, in W. Levine (ed.), Handbook of Control, CRC Press, 1996, 1339–1350.

[9] Awcock, G. W., R. Thomas. Applied Image Processing, New York, McGraw Hill, 1995.

[10] Ranky, P. G. Advanced Machine Vision Systems and Application Examples. *Sensor Review*, **23** (2003), No. **3**, 242–245.

[11] Mathworks, Image Acquisition Toolbox for Use with MATLAB User's Guide, The Mathworks Inc., 2005.

[12] Mathworks, Image Processing Toolbox for Use with MATLAB User's Guide, The Mathworks Inc., 2004.

[13] Jain, A. K. Fundamentals of Digital Image Processing, New Jersey, Prentice Hall, 1989.

[14] Rogers, D. F. Procedural Elements of Computer Graphics (2nd edition), New York, McGraw Hill, 1997.

[15] Gonzalez, R. C., R. E. Woods, S. L. Eddins. Digital Image Processing using MATLAB, New Jersey, Pearson Prentice Hall, 2004.