Sensors and Materials, Vol. 34, No. 10 (2022) 3895–3910
MYU Tokyo

S & M 3084

Development of Automated Optical Inspection and Classification Systems

Pu-Sheng Tsai,[1] Ter-Feng Wu,[2*] Jen-Yang Chen,[1] and Chia-Luen Tsai[3]

[1]Department of Electronic Engineering, Ming Chuan University, Taoyuan 333321, Taiwan (R.O.C.)
[2]Department of Electrical Engineering, National Ilan University, Yilan 260007, Taiwan (R.O.C.)
[3]Department of Electronic Engineering, China University of Science and Technology, Taipei 115311, Taiwan (R.O.C.)

(Received July 4, 2022; accepted October 4, 2022)

Keywords: manipulator, machine vision, attitude control, optical detection, identification and classification

In this study, we propose a novel system for automatic optical inspection (AOI) by integrating the Dobot Magician four-axis manipulator and machine vision. In the proposed system, the attitude control for the end joint rotates the claw kit to clamp the workpiece moving on the conveyor belt and performs color classification. The system conducts feature detection by examining the workpiece from different angles through cameras by using the working platform Python. Furthermore, the system identifies the positions of detected features and the center point and inclination angle of the workpiece. Finally, the Dobot Magician manipulator moves to the appropriate clamping position to pick up the object and then moves the object to the specified position. In this study, objects are preliminarily classified on the conveyor belt by using a manipulator through a photoelectric switch and a color recognition sensor, and the Hough transform is used to calculate the center point and rotation angle of the object for image recognition. To integrate image processing and Dobot manipulator motion control to complete the automatic simulation platform, an AOI setup is built for a production line. The system cooperates with the Python extension kit Tkinter to establish a user-friendly human–machine interface environment, which promotes the development and applicability of the automatic production platform.

1. Introduction

Automatic optical inspection (AOI) is a form of noncontact measurement aimed at improving machining precision, finding product defects, enabling automatic analysis, and decision-making.(1) By integrating machine vision with motion control, the efficiency of production-line automation for intelligent manufacturing can be improved effectively. Visionatics Inc. offers a series of image-processing libraries for machine vision applications, such as VIVA Vision products for image preprocessing and enhancement, image object capture, template comparison, contour comparison, image measurement, image detection, and image recognition. AAEON

*Corresponding author: e-mail: tfwu@niu.edu.tw
https://doi.org/10.18494/SAM4029

ISSN 0914-4935
https://myukk.org/

Technology has developed an embedded host computer and products for machine vision.[2] Many researchers have achieved groundbreaking results in machine vision and robot arms for production automation systems. Six-axis robot arms have been combined with machine vision to detect features on printed circuit boards,[3] including drilling positions and solder paste defects.[4,5] An automatic optical detection system has been used to detect defects on electronic components such as semiconductor chips, flat panel displays, printed circuit boards, and LEDs.[6] To accommodate the development of the microelectronics industry, the size of surface-mount devices (SMDs) continues to be scaled down while the mounting density grows, thereby increasing the importance of the AOI of solder joints. A new surface profilometry technique has been proposed for profiling a wafer surface with both diffuse and specular reflective properties.[7] The proposed technique is effective for 3D microscale surface profilometry in *in situ* semiconductor automated optical inspection. Zhao *et al.* applied NI Vision (the National Instrument Vision Development Module) to the AOI system for SMDs and reported and discussed the inspection results.[8] Researchers have proposed machine vision to draw and design a robot arm mechanism to solve the rubik cube by SolidWorks, obtaining six colors and outlining the rubik cube through webcam image capture and MATLAB image processing.[9] Furthermore, they converted the captured information into code and controlled the motor using an Arduino microcontroller to repeatedly create the rubik cube. Liang *et al.* developed an improved sliding mode control (SMC) scheme for the position tracking of a robot manipulator with parameter uncertainties and external disturbances.[10] Su *et al.* proposed an alternative control scheme including a directional genetic algorithm controller (DGAC), and a supervisory controller was developed to control the position of an electrical servo drive in this study.[11] The position and shape of a target object were identified by using machine vision. Moreover, the shape features of the target were obtained by adjacent pixel analysis, and the target database was established with coordinate rotation. The method proposed enables a robot arm to determine the position of randomly placed target objects and selects the best clamping position according to the fixture size parameters and object centroid. [12]

The Hough transform (HT) has always played a very important role in the field of image processing. It perfectly combines image data with mathematical geometry and maps huge pixel data in the image space to the parameter space by mathematical deduction. The cumulative voting method is used to calculate the local maximum and obtain the characteristic parameters that conform to the specific curve. For example, the straight line needs a two-dimensional parameter space, including the slope and intercept. The classical HT is mainly used to detect whether there are straight lines in an image.[13,14] The research was gradually extended to the recognition of circles[15] and ellipses.[16] The classical HT votes in the multidimensional parameter space during the detection of straight lines, circles, and ellipses, which requires a huge memory space and a heavy computational burden. The random HT (RHT)[17] and gradient HT[18] are variants of the traditional HT, which were developed to reduce the computational load. In Refs. 19 and 20, the line detection technique based on the HT was applied to the warning of straight-line deviation for autonomous vehicle driving in a complex environment, and good results were obtained. The HT is a feasible and robust approach in many practical applications. In this paper, we propose an innovative algorithm that uses the HT to detect rectangular objects

on a conveyor belt. The center points and rotation angles of rectangular objects are calculated using an image processing algorithm that can be used to drive the Dobot Magician end-effect gripper to clamp rectangular objects.

Dobot Magician launched by Dobot Inc. is a multifunctional desktop robotic arm comprising four units:[21,22] a manipulator, an IO expansion module (gripper kit, air pump box, conveyor belt, photoelectric switch, and color recognition sensor), a Dobot user interface, and a machine vision unit. In terms of software, QT, C#, Python, Java, VB, and other application programming interfaces (APIs) are provided. In this paper, Python is used as the working platform, and the OpenCV image-processing development kit and Python API provided by Dobot Magician are used to control the manipulator. To verify the feasibility of applying the manipulator to production-line automation, we performed a laboratory simulation. We established a conveyor belt to transport finished products: products were picked from the conveyor belt by the manipulator and classified from different angles using a photoelectric switch, color recognition sensor, and manipulator. The HT was used to calculate the rotation angle of the object to control the rotation angle of the end joint of the manipulator to clamp the object smoothly.

The remainder of this paper is organized as follows. In Sect. 2, we describe the architecture of the proposed system and the specifications of Dobot Magician. The HT principle and rectangular object detection algorithm are provided in Sect. 3. In Sect. 4, we describe how Dobot Magician is driven to reach the specified position, and the angle of the end joint is rotated to grasp the object. In Sect. 5, we show the experimental results and describe the user interface created using Tkinter in a Python environment. In Sect. 6, we present our conclusions and future works.

2. System Architecture

2.1 Specifications of Dobot Magician

The robot module used in this study is Dobot Magician launched by Dobot. The main body comprises the base, base arm, rear arm, forearm, and end effector. The corresponding four joints are termed J1, J2, J3, and J4. The second joint (J2) is assigned as the origin of the Cartesian coordinates (0, 0, 0), while the coordinates of the end tool are defined as (x, y, z). The power supply and warning lights are located above the base, as shown in Fig. 1. The length of the forearm of the manipulator is 147 mm, the length of the rear arm is 135 mm, and the length of the base arm is 138 mm, as shown in Fig. 1.

The maximum load of the manipulator is 500 g, the maximum extension distance is 320 mm, the movement angle of the base arm joint J1 is (−90–+90°), the movement angle of the rear arm joint J2 is (0–+85°), the movement angle of the forearm joint J3 is (−10–+90°), and the movement angle of the end tool J4 is (−135–135°). The working range of the X–Y-axis of the manipulator is shown in Fig. 2(a), and the activity space of the Z-axis is shown in Fig. 2(b). In addition, under a load of 250 g, the maximum rotation speed of the front and rear arms and the base is 320 °/s, and the maximum rotation speed of the end tool is 320°/s.

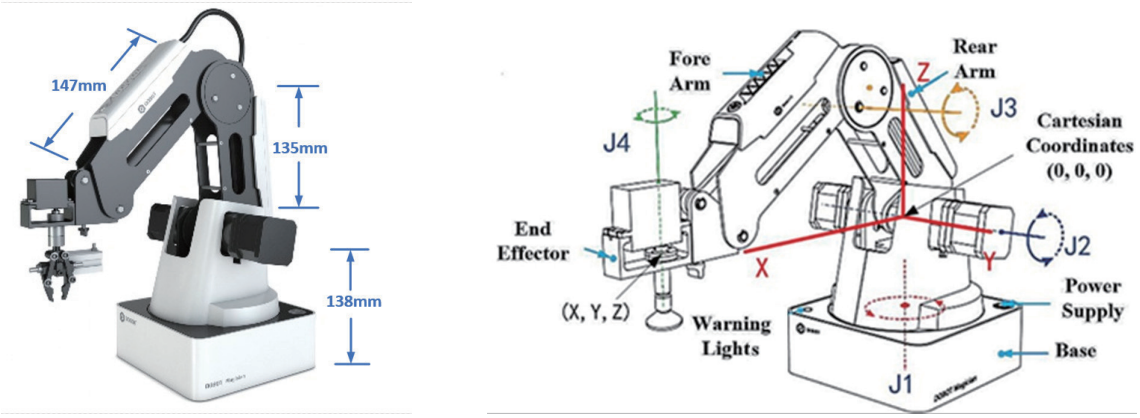
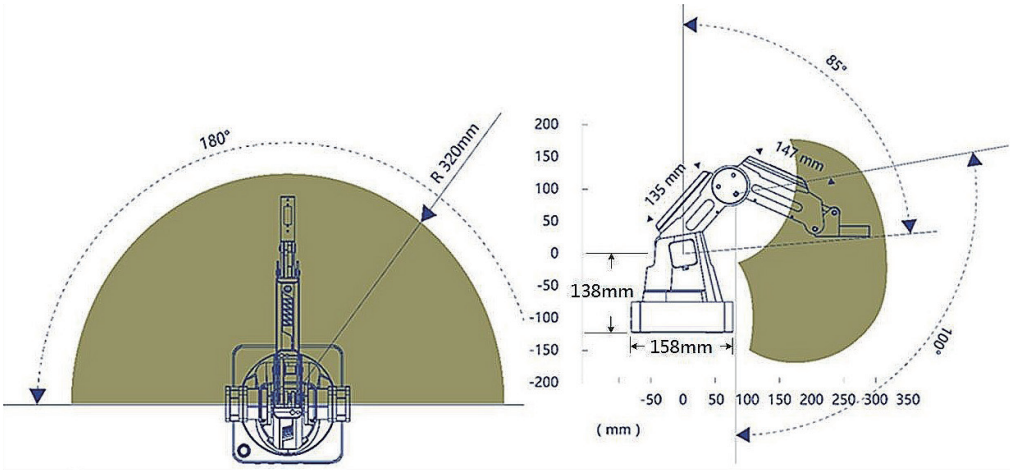


Fig. 1. (Color online) Dobot Magician.



(a) (b)

Fig. 2. (Color online) Working ranges of manipulator: (a) *X*–*Y*-axis and (b) *Z*-axis.

2.2 Dobot Magician expansion module

In this study, four expansion modules are applied: a gripper kit, air pump box, photoelectric switch, and color sensor. Figure 3 illustrates the installation of the gripper kit and air pump box. The gripper kit is installed with a 2.5 mm Allen wrench. Once installed, the signal line is connected to GP3, which is the terminal of the forearm. The air pump box pneumatically controls equipment such as suction cups and claws. It is installed by connecting an air pipe to the equipment and installing the signal and control lines to GP1 and SW1 at the I/O interface at the back of the base. The conveyor belt module drives the conveyor belt through the stepping motor, and the motor is driven through the four pins of Stepper 1, as shown in Fig. 3. A photoelectric switch is installed next to the conveyor belt, and the signal line is installed on GP2 at the base I/O interface.

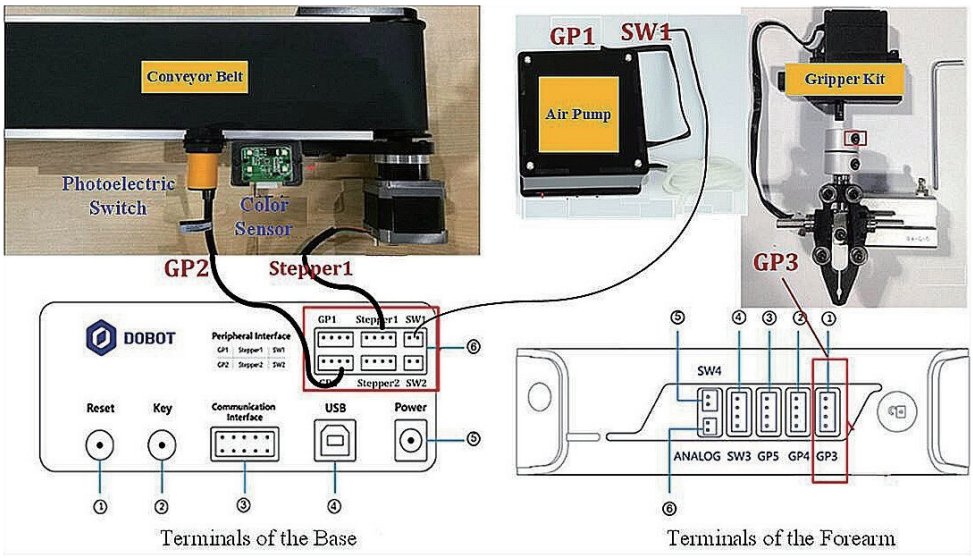


Fig. 3. (Color online) Installation of expansion modules.

3. Dobot Magician Expansion Module

The streaming image obtained by the video lens of LifeCam Studio installed on the fixture is processed at the user interface. System settings are shown in Fig. 4. We use Python language as the development platform and apply the HT to find the rotation angle of each moving object on the conveyor belt. This information allows the end joint of the manipulator to rotate at the same angle to grasp the object.

3.1 HT principle

The HT was first proposed by Paul Hough in 1962 and popularized by Duda and Hart in 1972.[14] It is a feature extraction technology for image processing that recognizes geometric shapes. The classical HT is used to detect straight lines, although it can be extended to recognize circles, ellipses, and even irregular shapes.

In practice, rectangular coordinate systems cannot be selected in the image space, because for any vertical line, such as $x = c$, the slope is infinite. We adopt polar coordinates (ρ, θ) in the Hesse normal form, as shown in Fig. 5. Here, ρ represents the vertical distance from the origin to this point and θ represents the angle between the x-axis and the vertical line. The following relationship can then be obtained as

$$\rho = \alpha \cos\theta = (x + \beta)\cos\theta = (x + y\tan\theta)\cos\theta = x\cos\theta + y\sin\theta. \quad (1)$$

Here, (ρ, θ) represents the characteristics of the line equation in the image space, where ρ is the vertical distance from the origin to this line and θ is the angle between the axis and the vertical

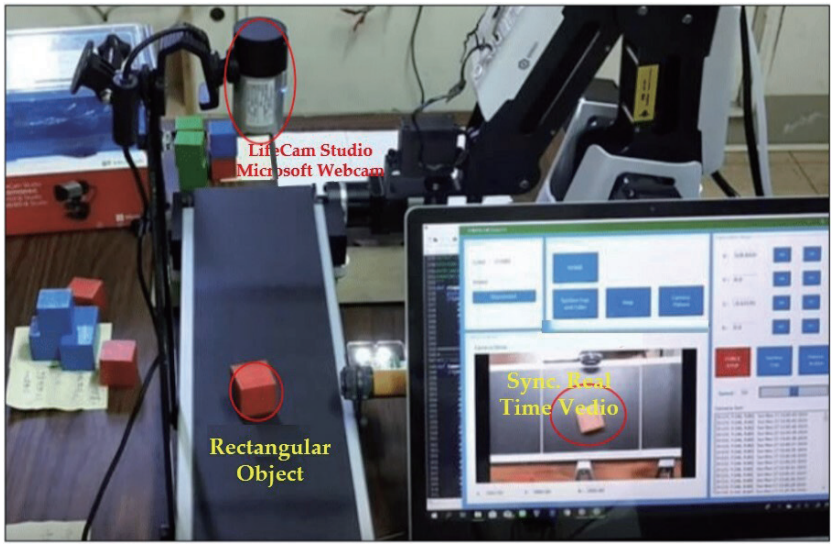


Fig. 4. (Color online) System architecture and hardware installation.

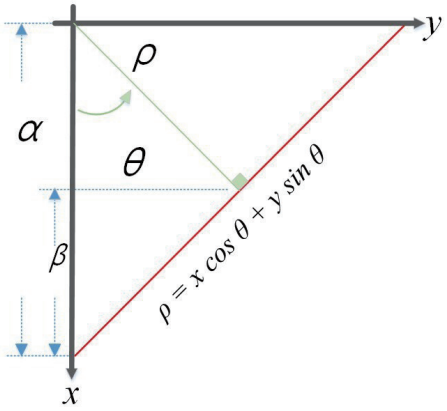


Fig. 5. (Color online) Representation of Hesse normal form in polar coordinates.

line. Subsequent instructions are in polar coordinates (ρ, θ). We present the procedure of the HT in detail in the following:

Step 1: First, establish the accumulator of the 2D array and initialize it by setting all elements in the array to 0. Figure 6 presents a schematic diagram of cumulative voting in the parameter space; for polar coordinates, the horizontal axis is θ and the longitudinal axis is ρ.

Step 2: Determine the size of the array, which directly affects the precision. If θ is accurate to 1°, 0–180 lines are required and the abscissa can reach 3.14 rad. To enable a general discussion, let n = 181. In general, to be accurate to the pixel level for ρ, the diagonal distance m must be used as the number of columns. In other words, the array size is $M \in R^{m \times n}$.

Step 3: Suppose there is a straight line on the image and its ends have pixel coordinates $A = (a_1, b_1)$, $B = (a_N, b_N)$, as shown in Fig. 7. Assume that N pixels are scanned by the camera

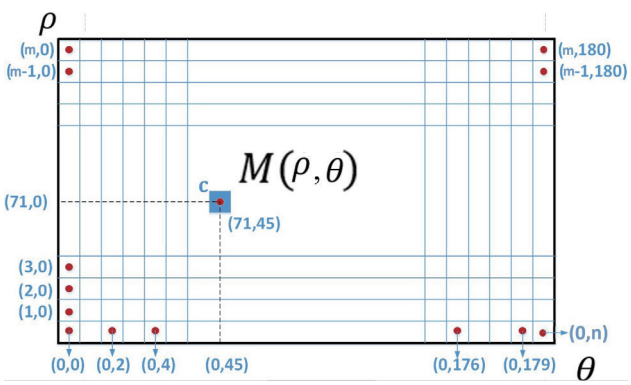


Fig. 6. (Color online) Accumulator voting array *M*.

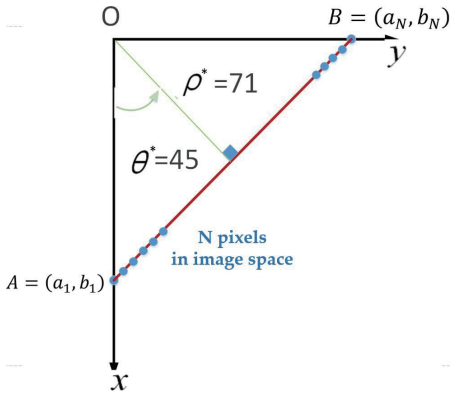


Fig. 7. (Color online) Straight line with *N* points.

from A to B and define these pixels as P_1 to P_N. Take the first point on the line $P_1 = A = (a_1, b_1)$ and substitute these x and y coordinates into Eq. (1) to obtain

$$L_1 : \rho = a_1 \cos\theta + b_1 \sin\theta. \tag{2}$$

Step 4: Substitute integer values of 0 to 180° for θ in Eq. (1) because θ is accurate to 1°. This returns a series of numerical pairs (ρ_0, 0), (ρ_1, 1), (ρ_2, 2), …, (ρ_{180}, 180). These 181 coordinates correspond to the accumulator voting array M. Then, add one to each corresponding position, as shown in Fig. 6. If there is a point in a series of numerical pairs (ρ^*, θ^*), then the accumulator voting array is set to $M(\rho^*, \theta^*) = 1$, where ρ^* is the number of pixels and θ^* is the number of degrees.

Step 5: Take all points on the line $P_1 = (a_1, b_1) - P_N = (a_N, b_N)$ and repeat steps 3 to 4 to update the accumulator voting array M. After a point on the straight line is operated on, one is added to the contents of the 181 positions in the accumulator array.

Step 6: Take the two end points as pixel coordinates $A = (a_1, b_1)$, $B = (a_N, b_N)$. The voting process is as follows:

(1) Total votes: $n \times N$, where $n = 181$.

(2) Candidates (accumulator array size): $m \times n$

(3) Count the contents of the accumulator voting array and find the maximum value:

$$\left(\rho^*, \theta^*\right) = \arg max\left(M\left(\rho, \theta\right)\right), \text{ where } 0 \le \rho \le m-1, 0 \le \theta < n-1. \tag{3}$$

(4) If $M(\rho^*, \theta^*)$ exceeds a specific threshold, a straight line exists in the image space. Furthermore, suppose that the coordinates of the peak value are $(\rho^*, \theta^*) = (71, 45)$, which means that the distance from the straight line to the origin is 71 pixels and the angle between its vertical line and the x-axis is 45°, as shown in Fig. 7.

According to the Hesse normal polar coordinate representation of Eq. (1), ρ is the vertical distance from the origin to the line and θ is the angle between the clockwise axis and the vertical

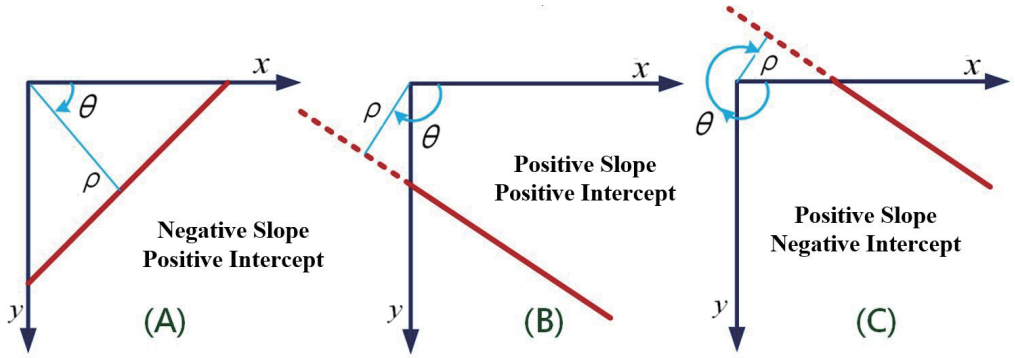


Fig. 8. (Color online) Possible distribution of lines in image space.

line. The first type of straight line is defined by $\rho > 0$ and $0° < \theta < 90°$ and the second type corresponds to $\rho > 0$ and $90° < \theta < 180°$. To make θ below 180°, we define the third type of straight line as $\rho < 0$ and $90° < \theta < 180°$. Thus, the third type of straight line is represented by $\theta > 180°$, as shown in Fig. 8.

3.2 Finding square rotation angle

The HT finds the center point and rotation angle of a square object in a color image, as shown in Figs. 9 and 10. The steps of image processing are as follows:

Step 1: Convert the HSV-filtered color image shown in Fig. 9(a) into the gray-scale image shown in Fig. 9(b).

Step 2: Binarize the gray-scale image to obtain the result shown in Fig. 9(c).

Step 3: Use Canny edge detection to obtain the coordinates of the image edge pixels, as shown in Fig. 9(d).

Step 4: Use the HT to map the pixels in the whole image to the parameter space ρ–θ, try to find the coordinates of the peak, and determine the polar coordinate parameters of the edge line segment of the square object (ρ_i, θ_i), $i = 1, 2, 3, 4, 5$, as shown in Fig. 10. Table 1 lists the five peak coordinates obtained from the running results of the program, indicating that there are five straight lines in the image space.

Using OpenCV, which is a cross-platform computer vision library, to draw the edge line segment of a square object in Python's platform, we can obtain the results presented in Figs. 9 and 10. Note that θ in Fig. 10 is the rotation angle of the square object.

Step 5: The effectiveness of detection using the HT is shown in Fig. 11. The HT can effectively detect the edge lines of the color part. The pixel coordinates (x_i, y_i), $i = 1, \ldots, n$ of the image edge are obtained as described in Step 3 using Canny edge detection. Here, the component x_i of each pixel is stored in a memory array X, and the component y_i of each pixel is stored in a memory array Y. The four vertex coordinates of a square, namely, (x_R, y_R), (x_L, y_L), (x_U, y_U), and (x_D, y_D), can be obtained by searching the X and Y arrays, where x_R and x_L represent the maximum and minimum in the array X, and y_U and y_D represent the maximum and minimum in array Y, respectively. inx_r and inx_l denote the index positions of the maximum and minimum

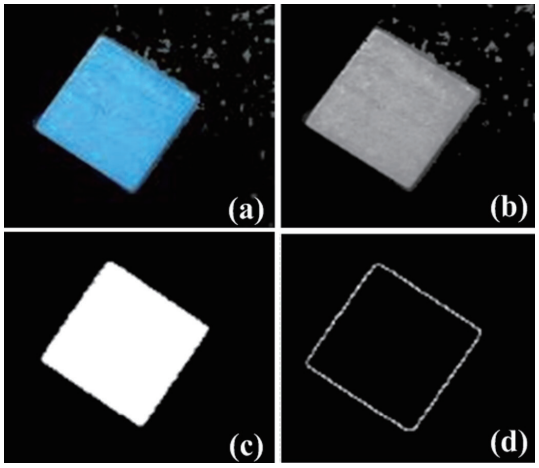


Fig. 9. (Color online) Image-processing program.

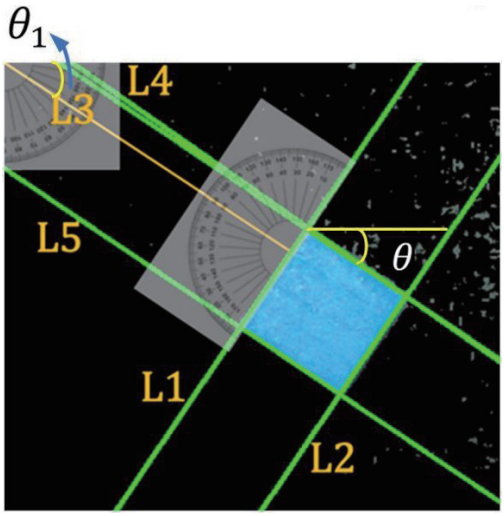


Fig. 10. (Color online) HT for L_1–L_5.

Table 1
HT detection results.

	L_1	L_2	L_3	L_4	L_5
ρ_i (pixel)	228	307	−19	−24	59
θ_i (angle)	34°	34°	124°	125°	124°

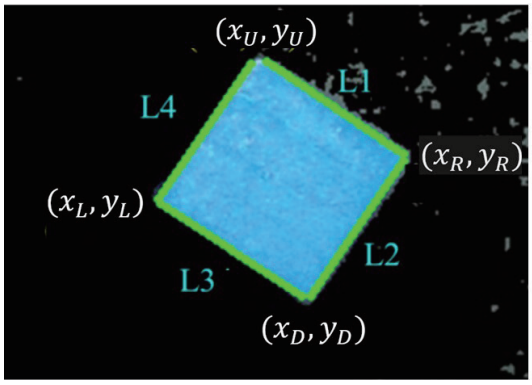


Fig. 11. (Color online) Finding object edge line using HT.

values located in array X, and iny_u and iny_d indicate the index positions of the maximum and minimum values located in array Y, respectively. The mathematical descriptions of square vertex coordinates are as follows:

$$max[X] = (x_R, inx_r);\ Y[inx_r] = y_R, \tag{4}$$

$$min[X] = (x_L, inx_l);\ Y[inx_l] = y_L, \tag{5}$$

$$max[Y] = (y_U, iny_u); X[iny_u] = x_U, \tag{6}$$

$$min[Y] = (y_D, iny_d); X[iny_d] = x_D. \tag{7}$$

In this paper, the coordinates of the center point of the object (x_c, y_c) are obtained as follows:

$$x_c = \left(\frac{x_R + x_L}{4} + \frac{x_U + x_D}{4}\right), y_c = \left(\frac{y_R + y_L}{4} + \frac{y_U + y_D}{4}\right). \tag{8}$$

Step 6: As shown in Fig. 12, the coordinates of the center point of the square object are easily obtained as (211, 166) using Eq. (8).

4. Control of Dobot Magician

The HT obtains the center point coordinates of square objects (x_c, y_c) on the conveyor belt, as shown in Eq. (8). It also determines the angle of rotation of the square object, as shown in Fig. 10. Next, the coordinates of the center point of the object are converted into the actual coordinates to drive the manipulator to reach the specified position and rotate the angle of the end joint to grasp the object.

4.1 Scale conversion

In this section, we confirm that the coordinates of the image are converted into the lens scale of the manipulator. We calculate the scale using the measurement experiment depicted in Fig. 13 and construct two vectors from the four-pixel vertex coordinates of the square shown in Fig. 11:

$$\boldsymbol{A} = (x_U - x_L, y_U - y_L), \boldsymbol{B} = (x_D - x_L, y_D - y_L). \tag{9}$$

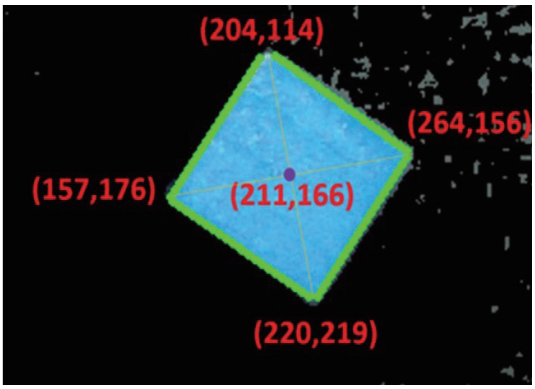


Fig. 12. (Color online) Calculating center point coordinates.

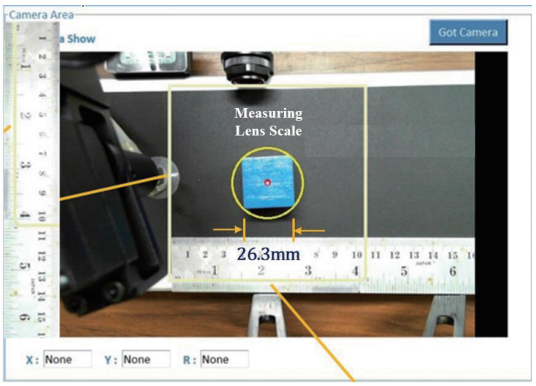


Fig. 13. (Color online) Calculation of scale.

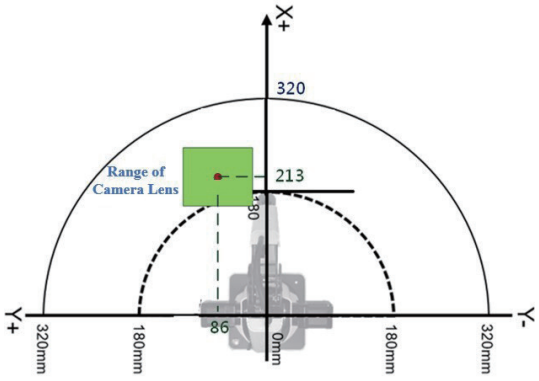


Fig. 14. (Color online) Camera position.

The area of the flat quadrilateral enclosed by vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ is

$$\Delta = \sqrt{|\boldsymbol{A}|^2 |\boldsymbol{B}|^2 - (\boldsymbol{A} \bullet \boldsymbol{B})^2}\,. \tag{10}$$

The values shown in Fig. 12 are substituted into Eq. (9) to obtain $\boldsymbol{A}$ and $\boldsymbol{B}$. Furthermore, we substitute the two vectors, $\boldsymbol{A}$ = (63, 43) and $\boldsymbol{B}$ = (47, −62), into Eq. (10) to obtain the area of the parallelogram enclosed by the two vectors as Δ = 5927 (pixels^2). In other words, the length of each side of the square is $\sqrt{\Delta} = 77$ (pixels). As shown in Fig. 13, the physical size is 26.3 mm. According to the measurement results, the scale relating the image and object coordinates is 1 pixels = 0.346 mm.

The driving mode of the manipulator is defined in the Cartesian coordinate system, as shown in Fig. 1. The origin (0, 0, 0) is assigned to joint J2, and the physical parameters (unit: mm) are input to drive the manipulator to the specified position. The green block in Fig. 14 is the range covered by the camera lens, and the position of the center point in Cartesian coordinates is (213.45, 85.97). To convert the coordinates (G_x, G_y) of the image to the Cartesian coordinates of the manipulator (D_x, D_y), we apply the following equations:

$$\begin{cases} D_x = 213.45 + G_y \times 0.346, \\ D_y = 85.97 + G_x \times 0.346. \end{cases} \tag{11}$$

4.1 Underlying communication protocol

Dobot Magician is provided with various APIs to assist with the control and expansion of the manipulator. The external microcontroller communicates with the manipulator through USB serial communication. The underlying communication protocol includes the header, data length, transmitted data, and check code of the data packet to ensure the correctness of data transmission. Table 2 presents the communication protocol adopted by Dobot Magician.

Table 2
Dobot Magician communication protocol.

Header	Len	Payload				Checksum
		ID	Ctrl		Params	
			rw	isQueued		
0xAA 0xAA	8 bit	8 bit	1 bit	1 bit	Command	Payload calculate

Table 3
Instructions for executing PTP displacement command.

Header	Len	Payload				Checksum
		ID	Ctrl		Params	
			rw	isQueued		
0xAA 0xAA	2 + 17	84	1	1 or 0	PTPCmd	Payload calculate

Taking the Dobot Magician point-to-point (PTP) displacement command as an example, Table 3 shows the format of the SetPTPCmd command, which specifies the setting parameters for the robot to move to Cartesian coordinates (x, y, z, r) = (250, 50, 0, 0).

5. Experimental Results

To verify the feasibility of applying the manipulator to production-line automation, we constructed an automatic optical detection and classification system in the laboratory with Python as the working platform combined with the OpenCV image-processing development kit and interface API. Table 4 shows the module kits loaded in the system.

5.1 User interface window

In the Python environment, the user interface is created using Tkinter and is based on the following four work blocks shown in Fig. 15: (1) serial communication connection area, (2) production process function, (3) manipulator control, and (4) image display processing area.

(1) Serial communication connection area: First, the link COM is selected to establish a connection between the user interface and the manipulator.
(2) Production process function: Keys enable users to quickly execute the automation process. The HOME key sets the system to zero calibration. The Suction Cup and Color key performs the function of classifying articles. The Stepper and Switch key drives the photoelectric switch and conveyor belt. The Camera Fixture key performs image processing for classification.
(3) Manipulator control: Control keys X, Y, Z, and R can add or subtract one unit coordinate parameter. The Force Stop key halts the arm action, the Suction Cup key starts the suction, the Fix Action key moves the fixture, and the Speed bar increases the speed and acceleration of the arm.
(4) Image display area: This area obtains real-time images of the lens and the calculated object pixel coordinates. The Camera key captures images during the operation of the image processing algorithm, including the original, gray-scale, and binary images, as in Fig. 15.

Table 4
Python configuration kit.

Based on Python				
System Kit sys time	User Interface Tkinter	Image Process OpenCV	Math Kit Numpy	Dobot API DobotDllType

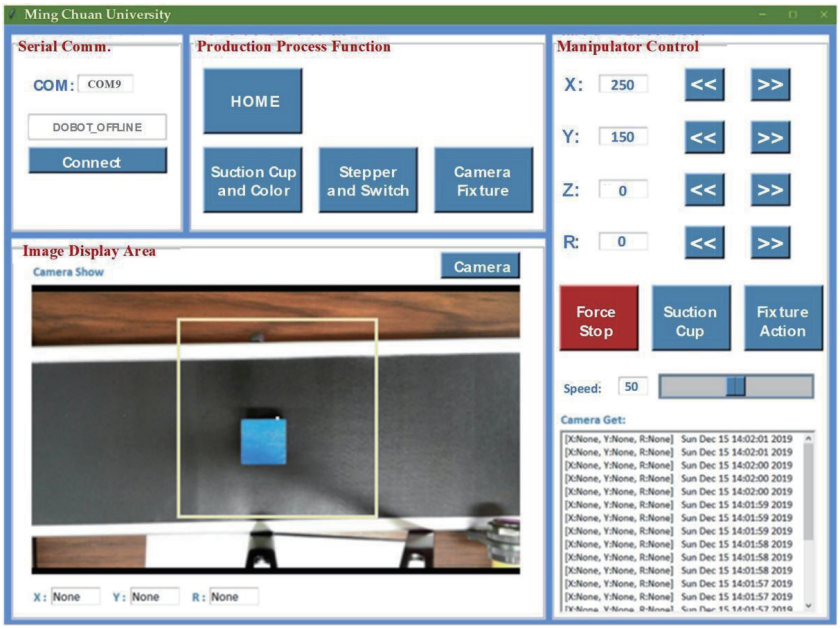


Fig. 15. (Color online) System user interface window.

Please refer to the following function test for the steps and procedures of the experiment.

[Function Test 1] Enter the production process function key block and click the Stepper and Switch button to drive the conveyor belt through the user interface.

[Function Test 2] When the red blocks reach the photoelectric switch, the conveyor belt stops. The red blocks are located at point A, as shown in Fig. 16.

[Function Test 3] Enter the production process function key block, click the Suction Cup and Color button to execute the color recognition algorithm, calculate the coordinates of the object at point A on the conveyor belt, and control the manipulator to move to point A to take the article through the user interface. The manipulator grasps the object with the suction cup, moves the object to the placement area with the PTP motion command, and places the object in one of three areas (red, green, or blue), as shown in Fig. 17.

[Function test 4] Enter the function key block of the production process, as shown in Fig. 18. Click the Camera Fixture button to initiate image processing algorithms to identify the color of the object, calculate the coordinates of the center point of the object, determine its inclination angle using the HT, and use the PTP motion command to control the manipulator to move to the coordinates of the center point of the object through the user interface. Rotate the end joint of the manipulator to match the inclination angle of the object, control the manipulator to grasp the object, move the manipulator to classify the object, and then place the object in one of three areas (red, green, or blue), as shown in Fig. 19.

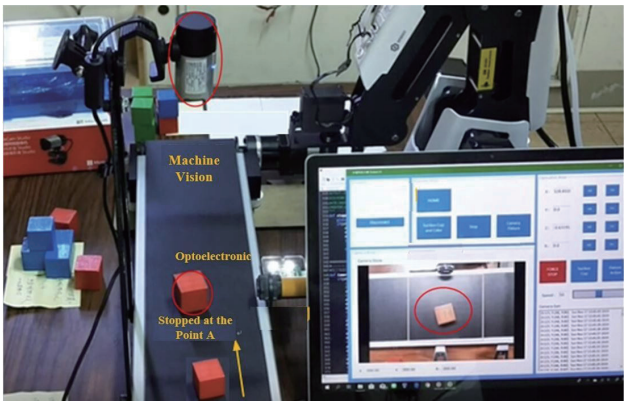


Fig. 16. (Color online) Red blocks stopping at point A

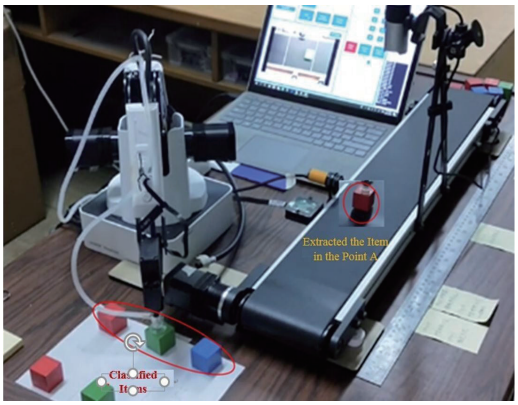


Fig. 17. (Color online) Suction cups picking blocks.

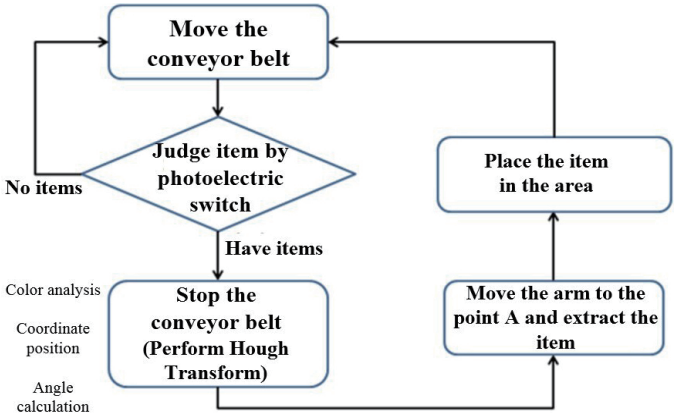


Fig. 18. (Color online) Test flow.

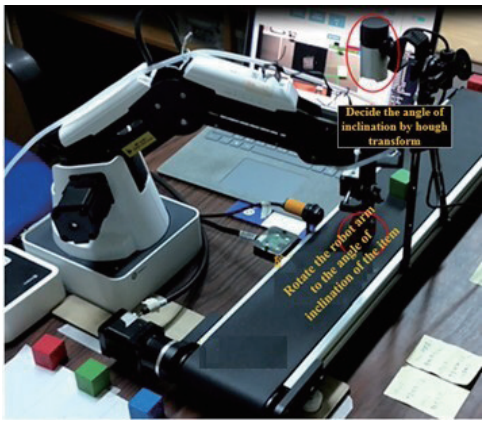


Fig. 19. (Color online) Picking end clamp of rotating manipulator.

6. Conclusion

In this study, we developed an AOI system using a multifunctional desktop Dobot Magician manipulator. The Dobot expansion kit includes a photoelectric switch, a color recognition sensor, an end connector of the conveyor belt, a suction cup, and a clamp to perform the action of object classification on the production line. In addition, we also expand mechanical vision to image recognition by using Python's package module, which calculates the center point and rotation angle of the object by the HT. The application of the HT is extended from line detection to square object detection. During the experiment, we used the Python package module Tkinter to design the graphical user interface (GUI). Therefore, this platform can be used to integrate image processing and manipulator control within an accessible human–machine interface to achieve AOI. Automated inspection improves the detection accuracy and speed, thereby speeding up the manufacturing process. The system is expected to make a valuable contribution to the field of smart manufacturing.

As a future direction for research, the machine vision unit can be extended with Raspberry Pi 3 expanding the GPIO interface and external cameras. The Raspberry Pi 3 adopts a Broadcom BCM2837 chipset, which is four-core 64-bit ARM Cortex-A8. The built-in Python language is used as the working platform to detect defective products such as deformation, shrinkage, or depression through image comparison. It further performs edge detection to find defective products with rough edges and cracks; the Dobot manipulator then rotates the end joint to pick the workpiece from the conveyor belt to remove defective products. In this paper, the HT is used to calculate the rotation angle of the object so that it can be grasped smoothly. The HT is a basic method for recognizing geometric shapes in image processing and is mainly used to separate geometric shapes with the same characteristics from images, such as line detection and circle detection. Therefore, the proposed system can also be applied to defect detection for circular or elliptical products.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology (MOST) of the Republic of China, Taiwan, for financially supporting this research under Contract No. MOST 111-2622-E-197-006 (https://www.most.gov.tw).

References

1 J. Beyerer, F. Puente León, and C. Frese: Machine Vision: Automated Visual Inspection Theory, Practice and Applications (Springer, Berlin, 2016).
2 AAEON Vision System Solution-Envision the Future of Industry 4.0: https://www.aaeon.com/en/ac/machine-vision (accessed August 2021).
3 M. Moganti, F. Ercal, C. H. Dagli, and S. Tsunekawa: Comput. Vision Image Understanding **63** (1996) 287. https://doi.org/10.1006/cviu.1996.0020
4 W. C. Wang, S. L. Chen, L. B. Chen, and W. J. Chang: IEEE Access **5** (2016) 10817. https://doi.org/10.1109/ACCESS.2016.2631658
5 J. Jiang, J. Cheng, and D. Tao: IEEE Trans. Compon. Packag. Manuf. Technol. **2** (2012) 1536. https://doi.org/10.1109/TCPMT.2012.2205149
6 A. A. R. M. A. Ebayyeh, and A. Mousavi: IEEE Access **8** (2020) 183192. https://doi.org/10.1109/ACCESS.2020.3029127
7 L. C. Chen, D. H. Duong, and C. S. Chen: Appl. Sci. **9** (2019) 2060. https://doi.org/10.3390/app9102060
8 H. Zhao, J. Cheng, and J. Jin: Proc. Int. Conf. Applied Superconductivity and Electromagnetic Devices (IEEE, 2009) 356–360.
9 Computer Vision and Rubik's Cube: http://kociemba.org/computervision.html (accessed August 2021).
10 Y. W. Liang, S. S.-D. Xu, and T. C. Chu: Proc. Int. Conf. Mechatronics and Automation (IEEE, 2007) 1593–1598.
11 K. H. Su and F. H. Hsiao: Adv. Mater. Res. **201–203** (2011) 2375. https://doi.org/10.4028/www.scientific.net/AMR.201-203.2375
12 Z. Shangguan, L. Wang, J. Zhang, and W. Don: Int. J. Aerosp. Eng. **2019** (2019) 1. https://doi.org/10.1155/2019/7050915
13 P. V. C. Hough: USA Patent No. 3069654 (1962). https://www.osti.gov/doepatents/biblio/4746348
14 R. O. Duda and P. E. Hart: Commun. ACM **15** (1972) 11. https://doi.org/10.1145/361237.361242
15 V. J. Schneider: Proc. SPIE 11736, Real-Time Image Processing and Deep Learning (SPIE, 2021) 201–214.
16 H. Fei, G. Yanling, and W. Lili: Proc. 2009 2nd Int. Conf. Information and Computing Science (ICIC, 2009) 301–305.
17 L. Xu, E. Oja, and P. Kultanen: Pattern Recognit. Lett. **11** (1990) 331. https://doi.org/10.1016/0167-8655(90)90042-Z

18 R. Cucchiara and F. Filicori: IEEE Trans. Pattern Anal. Mach. Intell. **20** (1998) 746. https://doi.org/10.1109/34.689304
19 Z. Zhang and X. Ma: J. Comput. Commun. **7** (2019) 65. https://doi.org/10.4236/jcc.2019.711005
20 J. G. Kuk, J. H. An, H. Ki, and N. I. Cho: Proc. 13th Int. IEEE Conf. Intelligent Transportation Systems (IEEE, 2010) 1344–1349.
21 M. R. Islam and M. A. Rahman: Proc. Int. Conf. Industrial Engineering and Operations Management (IEOM, 2019) 1507–1517.
22 Q. Ai, Q. Yang, M. Li, X. Feng, and W. Meng: Proc. 2018 10th Int. Conf. Measuring Technology and Mechatronics Automation (ICMTMA, 2018) 282–285.

About the Authors

Pu-Sheng Tsai was born in Taiwan, R.O.C., in 1962. He received his M.S. degree in automatic control from Feng Chia University, Taichung, Taiwan, R.O.C., in 1985 and his Ph.D. degree in electrical engineering from National Taiwan University, Taiwan, in 1998. He is currently an assistant professor in the Department of Electronic Engineering at Ming Chuan University, Taoyuan, Taiwan. His main research interests are in artificial intelligence, virtual reality somatosensory interaction, Internet of Things, and embedded microcontroller applications and design.

Ter-Feng Wu was born in Taiwan in 1962. He received his B.S. degree from the Department of Industrial Education, National Taiwan Normal University, Taipei, Taiwan, in 1986. He received his M.S. degree from the Department of Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1993. He received his Ph.D. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, in 2006. He is currently a professor and was the chair of the Department of Electrical Engineering, National Ilan University, Yilan, Taiwan, from 2009 to 2015. His research interests include automatic control, fuzzy CMAC, neural networks, unmanned aerial vehicles, green energy, and mobile robots.

Jen-Yang Chen was born in Hsin-Chu, Taiwan, R.O.C., in May 1960. He received his Ph.D. degree in electrical engineering from Tamkang University in 2000. He is currently a full professor in the Department of Electronic Engineering, Ming Chuan University. His research interests include intelligent control, soft computing, science education, robotic control, and adaptive control.

Chia-Luen Tsai was born in Taiwan, R.O.C., in 1981. He received his M.S. degree in electronic engineering from China University of Science and Technology, Taipei, Taiwan, R.O.C., in 2020. He currently works in KWAY Information Corporation as a senior software engineer. His main research interests are in embedded microcontroller applications, mechanical vision, human–machine interfaces, and artificial intelligence.