Vo Duy Cong
Lecturer
Ho Chi Minh City University of Technology(HCMUT)
Industrial Maintenance Training Center
268 Ly Thuong Kiet Street, District 10
Ho Chi Minh City
Vietnam

Le Duc Hanh
Doctor
Ho Chi Minh City University of Technology(HCMUT)
Faculty of Mechanical Engineering
268 Ly Thuong Kiet Street, District 10
Ho Chi Minh City
Vietnam

Le Hoai Phuong
Doctor
Ho Chi Minh City University of Technology (HCMUT)
Industrial Maintenance Training Center
268 Ly Thuong Kiet Street, District 10
Ho Chi Minh City
Vietnam

Dang Anh Duy
Lecturer
Ho Chi Minh City University of Technology (HCMUT)
Industrial Maintenance Training Center
268 Ly Thuong Kiet Street, District 10
Ho Chi Minh City
Vietnam

Design and Development of Robot Arm System for Classification and Sorting Using Machine Vision

The main focus of this paper is to design and develop a system of two robot arms for classifying and sorting objects based on shape and size using machine vision. The system uses a low-cost and high-performance hierarchical control system including one master and two slaves. Each slave is a robot controller based on a microcontroller that receives commands from the master to control the robot arm independently. The master is an embedded computer used for image processing, kinematic calculations, and communication. A simple and efficient image processing algorithm is proposed that can be implemented in real-time, helping to shorten the time of the sorting process. The proposed method uses a series of algorithms including contour finding, border extraction, centroid algorithm, and shape threshold to recognize objects and eliminate noise. The 3D coordinates of objects are estimated just by solving a linear equation system. Movements of the robot's joints are planned to follow a trapezoidal profile with the acceleration/deceleration phase, thus helping the robots move smoothly and reduce vibration. Experimental evaluation reveals the effectiveness and accuracy of the robotic vision system in the sorting process. The system can be used in the industrial process to reduce the required time to achieve the task of the production line, leading to improve the performance of the production line.

Keywords: *robot arm, computer vision, 3D localization, sorting, robot controller*

1. INTRODUCTION

Machine vision has become a critical component for many robot systems. The integration of vision technology has brought a variety of values such as improving accuracy, flexibility, and productivity, reducing labor costs and product damage, and expanding the application domain of robotics. A machine vision system consists of several essential components, which include vision sensors to interact with the environment, processing mechanism, and communication. The sensor in a machine vision is one or more cameras that capture images of objects and relay them to the processor for analysis. Vision processing employs algorithms to extract the required information, run the required inspection, and make a decision. Finally, the results are communicated to another device that logs or uses the information.

A vision system in the robotic system acquires and analyzes the image to extract the necessary information such as color and geometry of objects, segmentation of objects of interest, depth information, 3D coordinates of objects, … [1]. This information is used in the control process performed by a robot controller to control the movement of the robot and other components such as the sensors, motors, etc …

Received: August 2021, Accepted: December 2021
Correspondence to: Vo Duy Cong
Industrial Maintenance Training Center,
HCM City University of Technology
E-mail: congvd@hcmut.edu.vn
doi: 10.5937/fme2201181C

The robotic vision system is used in a variety of commercial and industrial applications, such as material inspection, object recognition, pattern recognition, assembly and disassembly, robot localization, vision guide robot, mapping, navigation, tracking, path planning, exploration, surveillance, search, recognition, inspection, …[2,26,27].

This paper develops a robotic vision system to automatically classify and sort objects based on their shape and size. The system consists of two robot arms for grasping objects, a conveyor belt for transporting objects, a camera for capturing an image of objects. Each robot is controlled by a microcontroller. The image processing, kinematics solving, and communication are implemented on an embedded computer. A series of image processing algorithms including contour finding, border extraction, centroid algorithm, and shape threshold are developed to detect, classify objects and find their position.

2. RELATED WORKS

In [3], Zhang et al. develop a machine vision system to automatically sort cherry tomatoes according to maturity. Three images of different angles are obtained from each cherry tomato and nine features were extracted from each image. Tomatoes are classified into three categories (unripe, half-ripe, and ripe) by using principal component analysis (PCA) and linear discrimination analysis (LDA) to analyze the features. Omid et al. [4] construct an experimental sorting system equipped with machine vision to sort tomatoes according to four quality criteria: maturity (color), defects, shape (oblong and circular), and size (small and large).

The software developed in this study evaluates tomato shape, size, maturity, and defect by its eccentricity, 2-D image area, mean color, and fullness parameter, respectively. An automatic apple sorting and quality inspection system is designed by Sofu et al. [5]. Their system consists of two identical industrial color cameras that are set on the roller conveyor to capture four images of any apple rolling on the conveyor. The images are analyzed to sort apples into different classes by their color, size and detect defective regions of the apples. The system also uses a load cell to measure the weight of apples. The proposed machine can sort an averagely of 432.000 apples per day with 79 sorting accuracy scores. Rafael et.al [6] develop a portable device based on a computer vision system for the automatic evaluation of green table olive quality in the field. The system consists of an illuminated cube that acquires images of fruit samples and generates an instantaneous report table. The external parameters of the report table consist of width, height, weight, color, Maturity Index, Bruising area, and Bruising Index.

A machine vision system can be used in the automatic counting system to automatically estimate the number of objects in a target area. It has been widely used in many fields [7][8][9]. In [10], Zhang et al. propose a fish counting method based on image density grading and local regression. In this paper, fish top-view images are divided into several connected area sub-images. Each sub-image is graded into different density levels by an area threshold and a backpropagation neural network (BPNN)-based regression model is constructed for each density-level dataset to count the number of fish. The experiment results show that the proposed method achieves a mean absolute error of 0.2985, a root means the square error of 0.6105, and a coefficient of determination of 0.9607. Tian et al. [11] propose a modified version of Counting Convolutional Neural Network in a fashion of end-to-end as a homogeneous, multi-branch architecture for pig counting. They combine both Counting CNN and ResNeXt in their deep learning architecture and tune a series of experimental parameters. The dataset used to train the CNN model is obtained from multiple websites and also captured from a real farm. After training is done, image patches are mapped to the corresponding density map and obtain the total number of pigs in the entire image by integrating the density map. Liping [12] proposes a novel end-to-end architecture based on Multi-Scale Adversarial Convolutional Neural Network (MSA-CNN) to generate crowd density and estimate the number of pedestrians in crowd images. The multi-column is used to extract high-dimensional features of the crowd image, and then a series of fractionally-strided convolutional layers is used to restore the detail of image features caused by max-pooling layers so that to improve the quality of the density map.

A robotic system equipped with a computer vision system can operate in an unstructured environment. The vision system recognizes the objects placed in the workspace and identifies the exact position of objects to lead the robot system. In [13], a robotic vision system that can operate in an unstructured environment is developed to recognize objects based on a high-performance Neural MUlticlassifier System (NEMUS). Various feature extraction methods (FEM) are applied to extract the feature sets as inputs for several classifiers. The outputs of all the classifiers are combined in a decision-making network (DM-Net) to perform the final classification task. The NEMUS is applied to a shape recognition task, under various levels of shape distortions and is also suitable for generic classification applications, such as shape discrimination, signal detection, and texture recognition.

A robotic vision system is presented in [14] to distinguish and sort object sort objects according to color and shape in real-time. A series of image processing techniques such as HSV threshold, shape properties, centroid algorithm, and border extraction is implemented to sort objects based on their color and shape. Then find the position of objects to pick and put the object on the right branch conveyor belt. Sangeetha et al. [15] use a stereo vision system to estimate the coordinate of targets and a three-DOF robotic arm is used to precisely position the target.

The kinematics algorithms and image processing are implemented in MATLAB 2012b and interface with NI6259 DAQ PCI card to control the movement of the robot arm. The results show that the arm reaches the target with a best-achieved accuracy of 2 cm. Also, in [16], a stereo vision system is developed to measure and predict the ball trajectory in real-time for a ping-pong robot. A multi-threshold segmentation algorithm is applied to detect the ball in the image.

The 3D position of the ball in the world coordinates is computed from two image coordinates by using the triangulation algorithm. Then, the flight trajectory of the ball is predicted using the aerodynamics model and rebound model. Aneesh et al. [17] design an efficient robot system that picks up the right colored and shaped objects and puts them down at the right place. The robot arm is controlled by a microcontroller. This system uses MATLAB for image processing to recognize the shape and uses a color sensor to recognize the color.

Tracking objects in real-time is becoming more and more important for some industrial tasks, such as grasping, sorting, and assembly, especially in a complex environment. The tracking is used in different fields as face tracking, color tracking, and shape tracking. The HSV spectrum is used to recognize objects based on shape and color to track a predefined object in real-time [18]. A stereo vision system extracts 3-D coordinates and a multiagent robot system is used for tracking, tooling, or handling operations [19].

The industrial tracking system is designed to provide tracking and sorting for products based on the shape and reject the products with low quality [20]. The robot manipulator can also track the trajectory using vision feedback [21]. The desired image trajectory is defined by a series of images are recorded when the engineer grasps the object and Model-free feedback-assisted iterative learning control strategy is used for repetitive tracking [22].

3. ROBOT SYSTEM DESCRIPTION AND DESIGN

3.1 Describe the operation of the system

This paper presents the design and implementation of a robotic vision system consisting of a conveyor to sort objects in real-time. Objects on the workspace are taken by a camera connected with a master controller. The master controller acquires and processes the image of objects to classification them depending on their size and shape. The master also determines the coordinates of objects in the workspace and converts them into joint coordinates by solving inverse kinematics equations. The system consists of two robot arms. The first robot receives joint angle values from the master to pick an object from the workspace and place it on the conveyor. The second robot controls the conveyor to transmit the object to sensor position, grasp and place the object in the right position. The movements of robots are point-to-point motion and the motions are planned by the robot controller. Figure 1 shows the basic components of the system and Figure 2 describes the steps of operation.

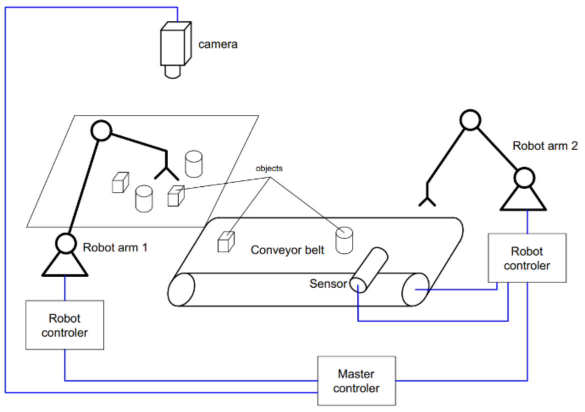


Figure 1. Basic components of the system

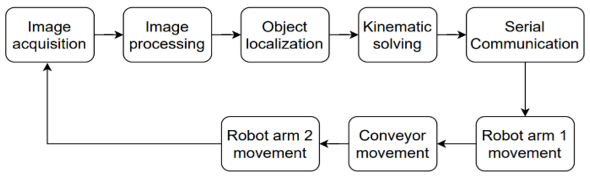


Figure 2. The steps of operation

3.2 Robot arm kinematics

Figure 3 shows the mechanical schematic of the robot arm. The end-effector of the robot is positioned using three parallelogram mechanisms that are composed of the following sets of links: {1,2,4,6},{0,1,3,5} and {5,6,7,8}.

For every motion of the robotic arm, the links 0, 5, and 8 are parallel to each other and thus keep the end-effector (link 8) always parallel to the horizontal. The robot has three degrees of freedom corresponding to the rotation angles θ_1, θ_2, and θ_3. Three stepper motors placed at each joint are used to control movements of links 0, 1, and 2, these motions are planned to create the desired motion of the end-effector. Before planning the trajectory, it is necessary to determine the value of the joint angles by solving the kinematics problem.

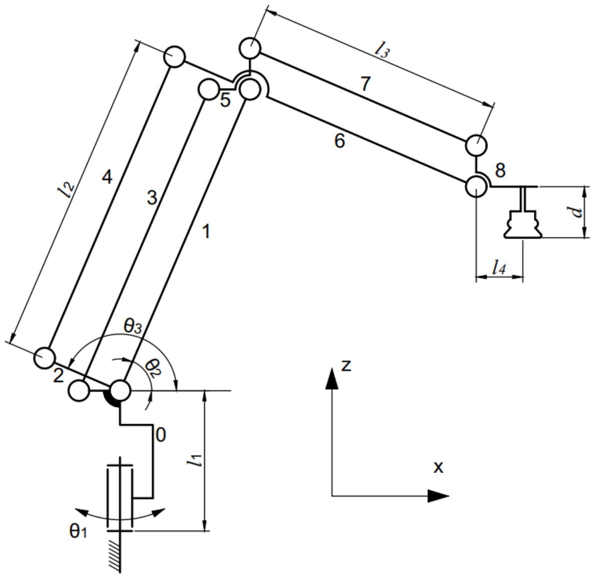


Figure 3. The robot arm schematic

From basic trigonometry, the position of the end effector can be written in terms of the joint angles as follows:

$$x = \cos\theta_1 \left(l_2 \cos\theta_2 - l_3 \cos\theta_3 + l_4\right) \tag{1}$$

$$y = \sin\theta_1 \left(l_2 \cos\theta_2 - l_3 \cos\theta_3 + l_4\right) \tag{2}$$

$$z = l_2 \sin\theta_2 - l_3 \sin\theta_3 + l_1 - d \tag{3}$$

Equations (1), (2), and (3) are called forward kinematic equations of the robot manipulator that describe the relationship between the end-effector coordinates and joint angles. To find the joint angles for a given set of end-effector coordinates, we need to solve the inverse kinematic equations.

From equations (1) and (2), we easily obtain:

$$\theta_1 = a\tan 2(y, x) \tag{4}$$

Here we use the atan2 function to get the unique joint angle θ_1. Square both sides in equations (1) and (2) then add them together:

$$l_2 \cos\theta_2 - l_3 \cos\theta_3 + l_4 = \pm\sqrt{x^2 + y^2} \tag{5}$$

Combine with equation (3) and group the unknowns on the left-hand side:

$$l_2 \cos\theta_2 - l_3 \cos\theta_3 = \pm\sqrt{x^2 + y^2} - l_4 = a \tag{6}$$

$$l_2 \sin\theta_2 - l_3 \sin\theta_3 = z + d - l_1 = b \tag{7}$$

Square both sides in each equation and add them together. After rearranging the terms, we get an equation in $\theta_3 - \theta_2$:

$$l_2^2 + l_3^2 - 2l_2l_3 \cos(\theta_3 - \theta_2) = a^2 + b^2 \tag{8}$$

Now, we can obtain the angle $\theta = \theta_3 - \theta_2$:

$$\theta = \theta_3 - \theta_2 = \pm\arccos\frac{l_2^2 + l_3^2 - a^2 - b^2}{2l_2l_3} \tag{9}$$

Rearrange the equation (7) according to the unknown angle θ_2, we get:

$$\left(l_2+l_3\cos\theta\right)\sin\theta_2-l_3\sin\theta\cos\theta_2=z+d-l_1=b \quad (10)$$

Define r and φ so that:

$$\begin{aligned} r&=\sqrt{\left(l_2+l_3\cos\theta\right)^2+\left(l_3\sin\theta\right)^2} \\ \sin\varphi&=\frac{l_2+l_3\cos\theta}{r} \\ \cos\varphi&=\frac{l_3\sin\theta}{r} \end{aligned} \quad (11)$$

The angle φ can be determined by using the atan2 function:

$$\varphi=a\tan 2\left(\frac{l_2+l_3\cos\theta}{r},\frac{l_3\sin\theta}{r}\right) \quad (12)$$

Substituting r and φ into (10) we get:

$$\cos\left(\theta_2+\varphi\right) \quad (13)$$

Finally, the solution of angle θ_2 is:

$$\theta_2=\pm\arccos\left(-\frac{b}{r}\right)-a\tan 2\left(\frac{l_2+l_3\cos\theta}{r},\frac{l_3\sin\theta}{r}\right) \quad (14)$$

There are four solutions for a given end-effector position. That means there are four configurations that the robot must choose to reach the desired position. The value of θ_1 is unique, the practical joint limits of joints 2 and 3 are used to get the unique configuration:

$$\begin{aligned} -\frac{\pi}{2}\le\theta_2\le\frac{\pi}{2} \\ 0\le\theta_3-\theta_2\le\pi \end{aligned} \quad (15)$$

3.3 Electronics design

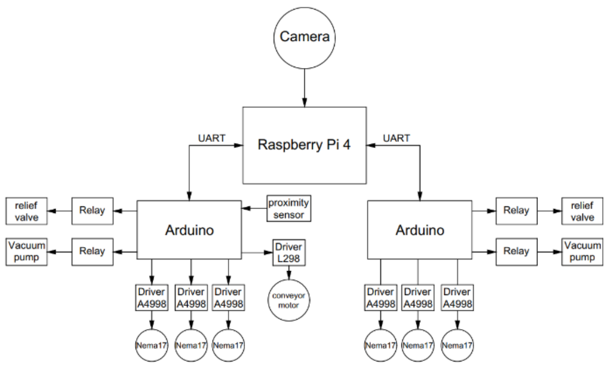


Figure 4. Electronic circuits block diagram

The control circuit of the system consists of one master circuit and two slave circuits as shown in Figure 4. The master controller is an embedded computer Raspberry Pi 4. This computer performs a variety of tasks that require high computational cost and process large amounts of data such as image processing, 3D localization, solving nonlinear kinematic equations, communication...Two slave circuits receive data from the master and control the movements of the robot and other components. Each slave is an Arduino board and is used as a simple robot controller. The Arduino board creates pulses and sends them to three A4998 drivers to drive stepper motors. In addition, the Arduino also outputs digital signals to control the solenoid valve, pump, and motor. Since the valve and pump operate at 12V, relays whose coils are energized by the 5V signal from the Arduino are used to turn ON and OFF them. The pump, solenoid valve, and vacuum suction cup are used in a vacuum system to grip and move objects. A proximity sensor is used to detect the object when it came to the picking up position. Arduino reads the signal from the proximity sensor to control the motor conveyor by outputting a signal to the L298 driver.

4. COMPUTER VISION SYSTEM

This section presents a method for shape and size classification and localization of objects with a simple algorithm and low computational time. The 3D coordinates of objects are estimated just by solving linear equations.

3.4 Image processing

Figure 5 shows the block diagram of the proposed method. The RGB image of objects taken by the camera is converted to a gray image and filters noise by a median filter.

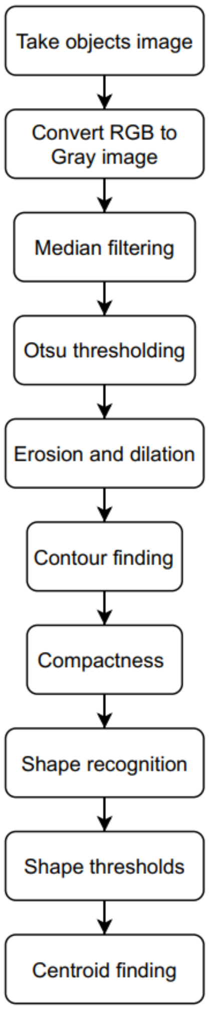


Figure 5. Block diagram of the image processing

The median filter is used to reduce “salt and pepper” noise and smooth away the edges. The idea of a median filter is to replace a pixel with the median value of the pixels in the M×M neighborhood. This paper uses 9×9 matrices

Then, the gray image is thresholded by Otsu's Binarization to extract the objects from their background. Otsu's Thresholding is an automatic global thresholding algorithm that selects a threshold automatically from a gray level histogram. The histogram image is separated into two clusters. The optimal threshold T is selected by the discriminant criterion. There are two options to find the threshold. The first is to minimize the within-class variance $\sigma_w^2(t)$ and the second is to maximize the between-class variance $\sigma_b^2(t)$:

$$\begin{aligned}\sigma_w^2 &= w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)\\ \sigma_b^2 &= w_1(t)w_2(t)\left(\mu_1(t) - \mu_2(t)\right)^2\end{aligned} \tag{16}$$

where $w_1(t)$, $w_2(t)$ are the probabilities of the two classes divided by a threshold$t$, σ_1, σ_2 are the variance and μ_1,μ_2 are the mean of each class.

In Fig. 6, by using Otsu's Thresholding, the binary image clearly shows the differences between the object and background. The background is marked with zero value while the objects are marked with one.

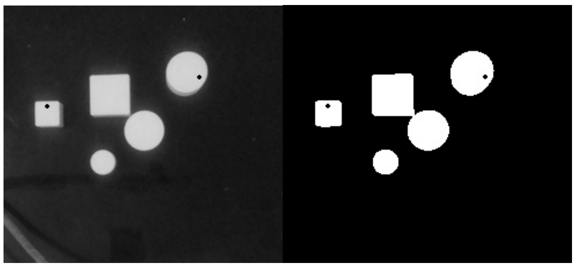


Figure 6. Otsu’s threshold

The morphological operators are applied to fill in small holes and eliminate small objects. Two basic morphological operators, dilation and erosion, are combined for specialized operations without changing the object size or shape. Firstly, a dilation followed by erosion is performed to fill holes in the objects while keeping the object sizes. Then, an erosion followed by dilation is applied to separate objects connected by a thin bridge of pixels and delete small noise objects.

The small red circles in Figure6 depict the "holes" and "thin bridge" in the input image. By implementing the morphological operators, these noise regions are eliminated.

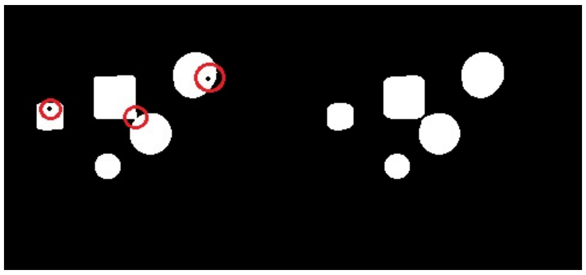


Figure 7. Morphological operators

Lastly, we find the contours of objects in the binary image and extract different features of contours, like area, perimeter, centroid… These features are used to classify objects and localization.

The shape of an object is recognized by computing the compactness:

$$c = \frac{p^2}{A} \tag{17}$$

where c is the compactness, p is the perimeter and $A$is the area. The perimeter is calculated by summing all pixels on the contour of the object. The area is equal to the zeroth-order image moment defined by:

$$\begin{aligned}M_{ij} &= \sum_u \sum_v u^i v^j I(u,v)\\ A = M_{00} &= \sum_u \sum_v I(u,v)\end{aligned} \tag{18}$$

where u and v are the row and column index, $I(u,v) = 1$ in the case of the binary image.

The centroid of the object in the image is given by the relations:

$$\begin{aligned}C_u &= \frac{M_{10}}{M_{00}}\\ C_v &= \frac{M_0 1}{M_{00}}\end{aligned} \tag{19}$$

The value of compactness$c$ and area A are used to classify objects. In this paper, objects are divided into four categories: small circle, large circle, small square, large square. The thresholds of c and A for classification are determined by experiment.

3.5 3D localization

After determining the centroid of the objects in the image according to equation (19). We can calculate the 3D coordinates of the object with the constraint that the height of the object is known in advance.

Define three coordinate frames F_w, F_r, and F_c corresponding to the world coordinate frame, the robot coordinate frame, and the camera coordinate system. The world coordinate frame is fixed at a known location. The robot coordinate frame is attached to the base of the robot and is also known. The camera coordinate frame is attached to the camera. The pose of a coordinate frame F_C relative to the world coordinate frame F_w can be represented as a homogeneous transformation T = [R t]. This homogeneous transformation is called the extrinsic parameters used to transform the world points to camera coordinates. The camera coordinates are mapped into the image plane using the intrinsic parameters:

$$\alpha p = K[R\,t]P \tag{20}$$

where $P = [X\ Y\ Z]^T$ and $p = [u\ v\ 1]^T$ are the coordinates of one point in the world frame and the image plane, respectively, α is a scale factor, K is the intrinsic matrix, R is the rotation matrix, t is the translation vector:

$$K=\begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$R=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, t=\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (21)$$

The camera calibration will estimate the intrinsic and extrinsic parameters. Substituting (21) into (20):

In this paper, the height of the object is fixed and known in advance and the centroid of the object in the image is calculated from the equation (19), so from equation (22), we can determine the coordinates X and Y of the object. Expand equation (22):

$$\alpha=\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}=\begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}\left(\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}+\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}\right) \quad (22)$$

$$a\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}=\begin{bmatrix} (fr_{11}+u_0r_{31})X+(fr_{12}+u_0r_{32})Y+(fr_{13}+u_0r_{33})Z+ft_x+u_0t_z \\ (fr_{21}+u_0r_{31})X+(fr_{22}+u_0r_{32})Y+(fr_{33}+u_0r_{33})Z+ft_y+v_0t_z \\ r_{31}X+r_{32}Y+r_{33}Z+t_z \end{bmatrix} \quad (23)$$

Denote:

$$a_1=fr_{11}+u_0r_{31};a_2=fr_{12}+u_0r_{32};a_3=(fr_{13}+u_0r_{33})Z+ft_x+u_0t_z$$
$$a_2=fr_{21}+u_0r_{31};b_2=fr_{22}+u_0r_{32};b_3=(fr_{13}+v_0r_{33})Z+ft_y+v_0t_z$$
$$c_1=r_{31};c_2=r_{32};c_3=r_{33}Z+t_z$$

The equations (23) are simplified:

$$u=\frac{a_1X+a_2Y+a_3}{c_1X+c_2Y+c_3}$$
$$u=\frac{b_1X+b_2Y+b_3}{c_1X+c_2Y+c_3} \quad (24)$$

Rewrite in terms of unknown X and Y coordinates:

$$\begin{cases}(uc_1-a_1)X+(uc_2-a_2)Y=a_3-uc_3 \\ (vc_1-b_1)X+(vc_2-b_2)Y=b_3-vc_3\end{cases} \quad (25)$$

Finally, X and Y coordinates can be easily obtained:

$$X=\frac{(a_3-uc_3)(vc_2-b_2)-(b_3-vc_3)(uc_2-a_2)}{(uc_1-a_1)(vc_2-b_2)-(vb_1-b_1)(uc_2-a_2)}$$
$$Y=\frac{(a_3-uc_3)(vc_1-b_1)-(b_3-vc_3)(uc_1-a_1)}{(uc_2-a_2)(vc_1-b_1)-(vb_2-b_2)(uc_1-a_1)} \quad (26)$$

The 3D coordinates of the object's centroid in the world frame are transformed to the robot frame using equation (27):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}=R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}+t \quad (27)$$

where $[x,y,z]^T$ and $[X, Y, Z]^T$ are the coordinates of the object in the robot frame and the world frame, respectively, R is the rotation matrix, t is the translation vector, representing the relationship between the two frames. These coordinates are converted to the joint angles using the inverse kinematics in section 3.2 as follows: solve the angle θ_1 from equation (4). Calculate coefficients a, b according to equations (6) and (7). Then, using these coefficients to calculate the angles θ_2 and θ_3 from equations (9) and (14).

5. TRAJECTORY PLANNING AND STEPPER MOTOR CONTROL

Trajectory planning creates reference signals for the robot controller so that the robot can move in the desired trajectory. Trajectory planning can be done either in the joint space or in the Cartesian space [23]. Using Joint Space Trajectories has many advantages such as less computation, easier to plan trajectories in real-time, and no problem with singularities. For pick and place applications in industrial, joint space trajectories are usually used.

The planning algorithm generates a function *q(t)* interpolating the given vectors of joint variables at each joint. In industrial practice, a trapezoidal velocity profile is usually assigned (see Figure 8. a). The velocity graph consists of three phases namely constant acceleration, constant velocity, and constant deceleration. Assume that the angle q_f from the initial position to the final position, the maximum speed ω_{max}, and the constant acceleration/deceleration $\dot{\omega}$ is given in advance. We need to determine the acceleration/deceleration time t_c, the time in the constant velocity phase t_v, the total time T, and the function of *q(t)*.

The velocity at the end of the acceleration phase is equal to the constant velocity, so:

$$t_c=\frac{\omega_{\max}}{\dot{\omega}} \quad (28)$$

And the angle after the acceleration is:

$$q_c=\frac{1}{2}\dot{\omega}t_c^2=\frac{\omega_{\max}^2}{2\dot{\omega}} \quad (29)$$

The area of the trapezoid is equal to the total angle q_f, so:

$$(t_c+t_v)\omega_{\max}=q_f \quad (30)$$

Therefore, the time of the constant velocity phase is:

$$t_v = \frac{q_f}{\omega_{\max}} - \frac{\omega_{\max}}{\dot{\omega}} \tag{31}$$

If $t_v > 0$ or $q_f \cdot \dot{\omega} > \omega_{\max}^2$, the velocity profile is a trapezoid, the total time is:

$$T = 2t_c + t_v = \frac{\omega_{\max}}{\dot{\omega}} + \frac{q_f}{\omega_{\max}} \tag{32}$$

The trajectory is formed by a linear segment connected by two parabolic segments:

$$q(t) = \begin{cases} \frac{1}{2}\dot{\omega}t^2 & 0 \le t \le t_c \\ \frac{\omega_{\max}^2}{2\dot{\omega}} + \omega_{\max}(t - t_c) & t_c \le t \le T - t_c \\ q_f - \frac{1}{2}\dot{\omega}(T-t)^2 & T - t_c \le t \le T \end{cases} \tag{33}$$

If $t_v < 0$ or $q_f \cdot \dot{\omega} > \omega_{\max}^2$, the velocity profile is a triangle that only consists of acceleration and deceleration (see Figure 8.b). The trajectory is formed by two parabolic segments, we have:

$$\frac{q_f}{2} = q_c = \frac{1}{2}\dot{\omega}t_c^2 \tag{34}$$

Therefore, the time of the acceleration/deceleration and total time is:

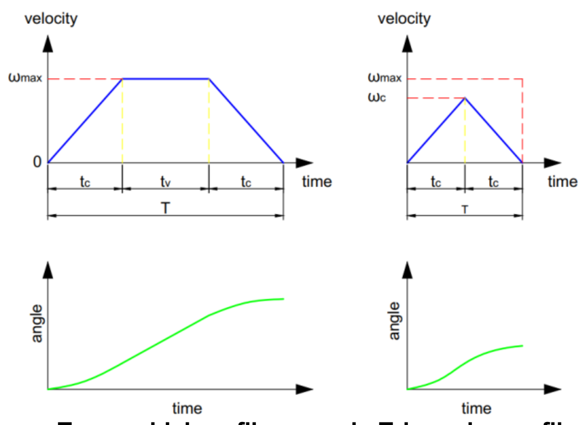
$$\begin{aligned} t_c &= \sqrt{\frac{q_f}{\dot{\omega}}} \\ T = 2t_c &= 2\sqrt{\frac{q_f}{\dot{\omega}}} \end{aligned} \tag{35}$$

The maximum velocity in this case is:

$$\omega_c = \dot{\omega}t_c = \sqrt{q_f\dot{\omega}} \le \omega_{\max} \tag{36}$$

The function of angle in the term of time t:

$$q(t) = \begin{cases} \frac{1}{2}\dot{\omega}t^2 & 0 \le t \le t_c \\ q_f - \frac{1}{2}\dot{\omega}(T-t)^2 & t_c \le t \le T \end{cases} \tag{37}$$



a. Trapezoidal profile **b. Triangular profile**

Figure 8. Velocity profile

A stepper motor is controlled by sending pulses to the motor driver. One pulse makes the motor rotate one constant step angle α. The change of speed is achieved by changing the time interval between successive steps. It is difficult to generate pulses if the velocity is variable because the time interval between two adjacent pulses is changed. Consider the constant acceleration phase, the joint angle for the nth step pulse is:

$$q_n = n\alpha = \frac{1}{2}\dot{\omega}t_n^2 \tag{38}$$

where n ≥ 0 is the step number, t_n is the time for the nth step pulse. The time interval between two adjacent pulses is:

$$\delta t_n = t_{n+1} - t_n = \sqrt{\frac{2(n+1)\alpha}{\dot{\omega}}} - \sqrt{\frac{2n\alpha}{\dot{\omega}}} = \sqrt{\frac{2\alpha}{\dot{\omega}}}\left(\sqrt{n+1} - \sqrt{n}\right) \tag{39}$$

It can be seen the time interval δt_n between two adjacent pulses is not linear and complex to calculate in real-time (calculating two square roots is time-consuming) for a mid-range microcontroller. Therefore, we use an approximation with less computational complexity (implemented by D. Austin [24]):

$$\begin{Bmatrix} \delta t_n = \delta t_{n-1} - \frac{2\delta t_{n-1}}{4n+1}, n > 0 \\ \delta t_0 = \sqrt{\frac{2\alpha}{\dot{\omega}}} \end{Bmatrix} \tag{40}$$

Motor step signals are generated by a 16-bit timer /counter module in the Arduino running at the frequency f. The delay δt programmed by the counter c is:

$$\delta t = \frac{c}{f} \tag{41}$$

Substituting into (39):

$$\begin{cases} c_n = c_{n-1} - \frac{2c_{n-1}}{4n+1}, n > 0 \\ c_0 = f\sqrt{\frac{2\alpha}{\dot{\omega}}} \end{cases} \tag{42}$$

This approximation introduces an error of 0.44 at n=1. There are two ways to compensate for this error: multiplying c_0 with 0.676 or using $c_1 = 0.4056c_0$ [23].

We can calculate the number of steps on acceleration phase by dividing the angle by step angle:

$$N_c = \frac{q_c}{\alpha} = \frac{\dot{\omega}t_c^2}{2\alpha} \tag{43}$$

where t_c is determined by equation (28) or (35). The acceleration stops when the number of steps n is equal to N_c. After that, the constant velocity phase is started. The timer delay in this phase is constant, so the value of the counter is:

$$c_n = f\frac{\alpha}{\omega_{\max}} \tag{44}$$

The stepper motor is kept at constant speed until the number of pulses reaches the value:

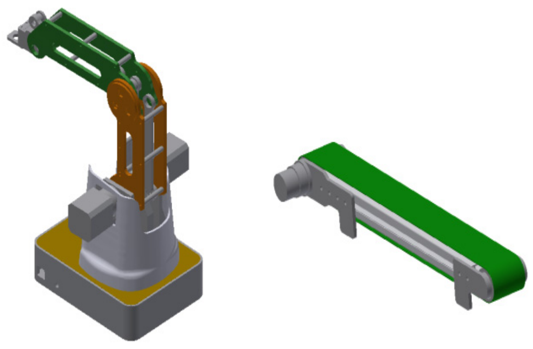
$$N_v = \frac{q_f - 2q_c}{\alpha} \quad (45)$$

Finally, deceleration starts. Equation (46) can be used to ramp the speed down to zero in the final step of a move of N_c steps (D. Austin [24]):

$$c_n = c_{n-1} + \frac{2c_{n-1}}{4(n - N_a) + 1}, n < N_c \quad (46)$$

6. RESULTS AND DISCUSSIONS

The mechanical models of the system are designed using the Autodesk Inventor software (see Figure 9). Separate parts of the models are saved in the stereolithography (STL) file format. Then the parts are fabricated by a 3D printer. Finally, the 3D printed components are assembled and with other parts (e.g., motors, bearings, shafts, aluminum frame, …) to produce a complete system as shown in Figure 10.



a. Robotic-arm model b. Conveyor model

Figure 9. Mechanical design on the Autodesk Inventor software

The kinematic dimensions of the robotic arm are as follows:
l_1 = 153.3 mm, l_2 = 135 mm, l_3 = 160 mm, l_4 = 45 mm, d = 25 mm. The joints of the robot arm are driven by 5.17:1 planetary gearbox stepper motors. The stepper motor can provide a maximum holding torque of 0.25 Nm, resulting in a maximum robot's payload of 500g. In addition, the motors use a maximum of only 10W of power each, and three motors combined use a maximum of only 30W, resulting in significant energy savings. Table 1 shows the specifications of the robotic arm.

The vision system uses a Raspberry Pi Camera Module with a Sony IMX219 8-megapixel sensor. This camera has a focal length of 3.04mm, an angle of view (diagonal) of 62.2 degrees, and a resolution of up to 3280 x 2464 pixels. In the project, we only use images with a resolution of 640x480 pixels to achieve faster processing speed.

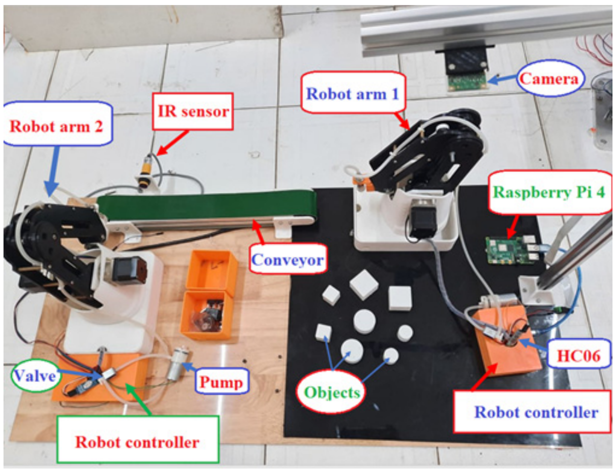


Figure 10. The experiment system

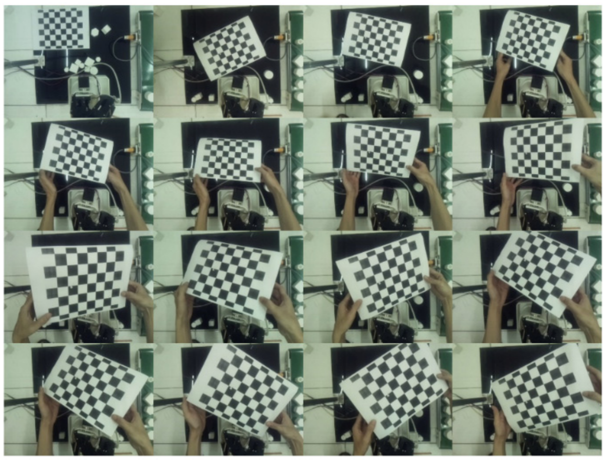


Figure 11. The chessboard images for calibration

Table 1. Specifications of the robotic arm.

Specifications	Number of Axis		3
	Payload		500g
	Max. Reach		340mm
	Communication		USB/Bluetooth
	Power Supply		110V/220V, 50/60Hhz
	Power In		12V/3A
	Consumption		32W Max
Axis Movement	Axis	Range	Max speed
	Joint 1 base	-90^0 to $+90^0$	320^0/s
	Joint 2 rear arm	-30^0 to 85^0	210^0/s
	Joint 3 forearm	-50^0 to 60^0	210^0/s
Physical	Weight		2.6kg
	Base Dimension		200mm x 150mm
	Materials		PLA plastic
	Controller		Arduino

A chessboard is employed to calibrate the parameters of the camera. The calibration uses the Camera Calibration Toolbox for Matlab® [25] Based on a total of 16 images of a planar chessboard (see Figure 11). The intrinsic and extrinsic parameters after calibration are as follows:

$$K = \begin{bmatrix} 492.78 & 0 & 322.17 \\ 0 & 494.30 & 234.55 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.9987 & 0.0319 & 0.0400 \\ -0.0292 & 0.9974 & -0.0659 \\ -0.0420 & 0.0647 & 0.9970 \end{bmatrix}; t = \begin{bmatrix} -186.57 \\ -247.26 \\ 609.73 \end{bmatrix}$$

Objects are classified into four types: small circle (sc), large circle (lc), small square (ss), large square (ls). By experiment, the thresholds of c and A to classify objects are as follows:

$$small\,circle: \begin{cases} 12 < c < 14 \\ 300 \le A \le 500 \end{cases}$$

$$large\,circle: \begin{cases} 12 < c < 14 \\ 850 \le A \le 1100 \end{cases}$$

$$small\,square: \begin{cases} 14 < c < 16 \\ 400 \le A \le 600 \end{cases}$$

$$small\,square: \begin{cases} 14 < c < 16 \\ 1050 \le A \le 1450 \end{cases}$$

The proposed method is tested on a database consisting of 100 images. This database can be divided into five groups which are dataset that contains only one object, two objects, three objects, four objects, multiple objects with noise objects. Figure 12 illustrates the categories in the dataset and Table 2 shows the corresponding results.

Table 2. The accuracy of the classification method

Categories	Number of images	Accuracy %
One object	16	100
Two objects	25	100
Three objects	20	95
Four objects	25	88
Noise object	14	85

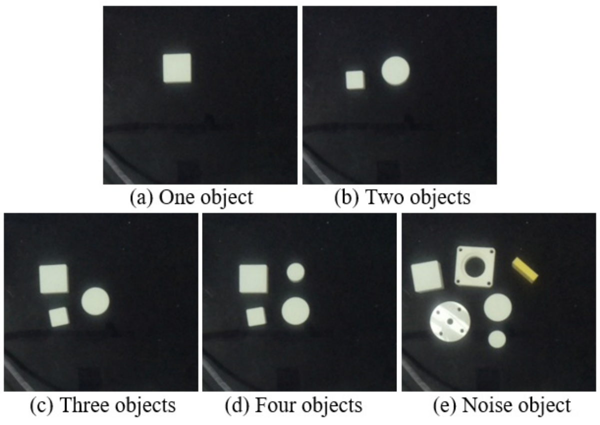


Figure 12. Five categorizes dataset

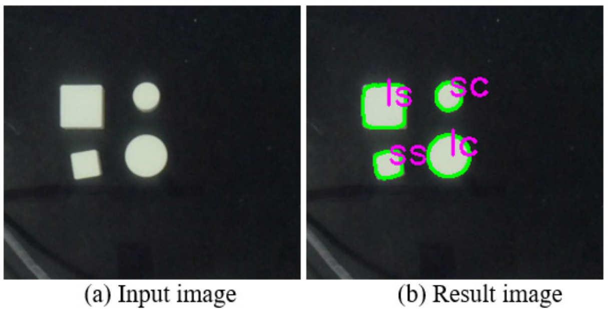


Figure 13. Example of the successful result

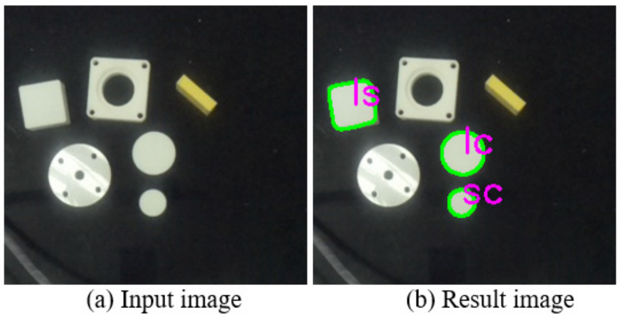


Figure 14. Example of the successful result with noise objects

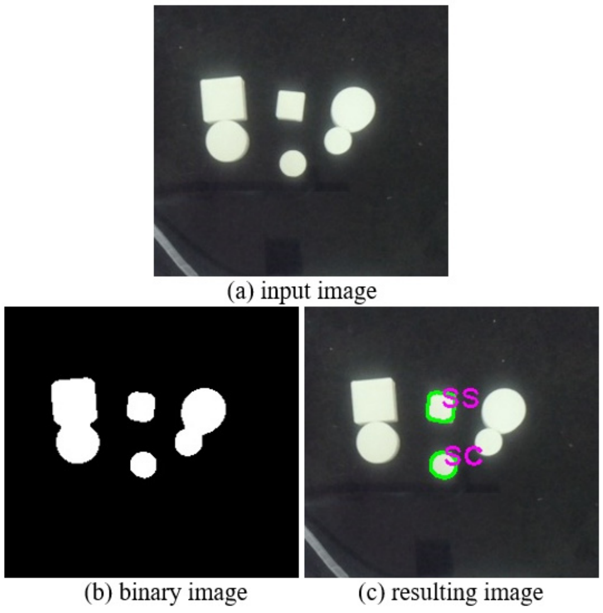


Figure 15. Example of the incorrect result

Figure 13 shows the example of successful detection and classification by using the proposed method. In the resulting image, the objects are labeled according to their shape and size. Figure 14 demonstrates that the proposed method can remove noise objects. The noise objects have the desired shape but the size is different from the object of interest. Figure 15 shows a case of incorrect detection. The reasons for the faulty recognition may be due to lighting conditions, resulting in the image is not thresholded properly or some objects being very close to each other that cannot be separated by using the morphological operators.

After detecting the object, the 2D centroid of the object is calculated and the 3D coordinates are estimated using the equations in section 4.2. From the above image dataset, the 3D coordinates of the objects are calculated and compared to the coordinates measured directly using a ruler. Then, the errors are extracted. The results show that the estimation method based on vision has an average error of 3.47 mm. Table 3 shows the first 20 results.

Table 3. The 3D estimated coordinates and errors

No.	**Image**(pixel)		**Estimate**(mm)		**Real**(mm)		**Error** (mm)
	cx	cy	X	Y	X	Y	
1	338	209	184.59	-5.8	183	-7	1.99
2	375	181	216.13	-50.42	218	-50	1.92
3	289	166	237.19	50.78	239	48	3.32
4	344	150	253.59	-14.92	251	-16	2.81
5	344	149	254.76	-14.96	254	-14	1.22
6	317	165	237.21	17.5	240	16	3.17
7	369	153	249.05	-44.2	251	-43	2.29
8	321	153	248.73	12.38	245	15	4.56
9	375	154	247.63	-51.2	246	-49	2.74
10	320	203	192.36	15.38	195	18	3.72
11	367	183	214.10	-40.94	214	-40	0.94
12	313	154	250.24	21.81	251	23	1.41
13	318	202	193.62	17.72	196	13	5.29
14	369	195	199.93	-42.94	204	-46	5.09
15	315	154	250.16	19.44	247	22	4.07
16	369	145	258.34	-44.44	258	-41	3.46
17	372	232	156.11	-45.37	155	-46	1.28
18	373	191	204.48	-47.77	200	-48	4.48
19	282	151	255.07	58.46	257	55	3.96
20	407	143	259.03	-88.86	260	-87	2.1

As soon as the analysis of the image is completed, the coordinates are sent to the first robot arm to pick and place the object on the conveyor, and at the same time, the type of the object is also sent to the second robot. After the first robot arm completes the pick and place operation, the next image is requested to continue the process, and the robot backs to the initial position, waiting for the new command from the master. The conveyor transports the object to the sensor position. Then the conveyor stops, the second robot picks and places the object in the right position according to the command received from the master.

During the transporting and sorting of the object by the second robot, the processing on the master is also performed. So, the time of the process is minimal. The system needs a total time of 2.2 to 2.4s to achieve the sorting of one object. This time is also equal to the time it takes the first robot to pick and place the object on the conveyor because the time to transport and sort the object by the second robot is only about 2.1s.

Figure 16 shows the rotation angles of the robot's joints when applying the trapezoidal velocity profile. The three joints have the same acceleration and maximum speed, so the acceleration and deceleration times are the same, but the time of the constant velocity phase is different because the rotation angle at each joint is not the same. Using the trapezoidal profile with the acceleration/deceleration phase helps the robot move smoothly and reduce vibration.

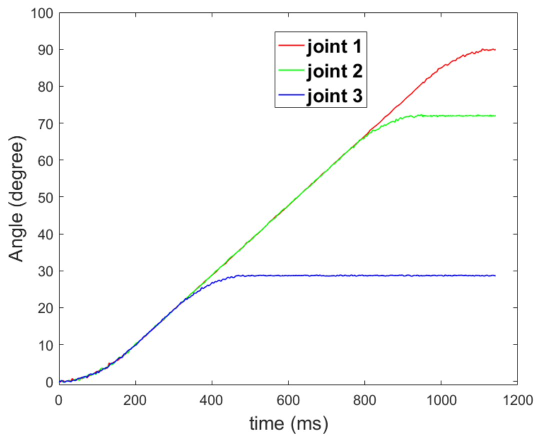


Figure 16. The rotation angles of the robot's joints when applying the trapezoidal velocity profiles

7. CONCLUSIONS

In this paper, the robotic vision system is designed to sort objects according to their size and shape. The proposed system can be used in the industrial process to reduce the required time to achieve the task of the production line, leading to improve the performance of the production line.

The movements of two robotic arms are driven by stepper motors and are controlled by the Arduino board. An approximation with less computational complexity is used to approximate the trapezoidal velocity profile that can be implemented on the Arduino controller. This helps the robot can move smoothly and reduce vibration. The Raspberry Pi computer is used as a master controller to control and communicate between two robots. The master is used for image processing, kinematic calculations, and communication. The combination of the two controller boards results in high performance and a low development cost.

The vision system is a key component in the sorting process. The accuracy and performance of the vision system directly affect the performance and speed of the sorting process. A simple and efficient image processing algorithm is been proposed that can be implemented in real-time, helping to shorten the time of the sorting process. The proposed method uses a series of algorithms including contour finding, border extraction, centroid algorithm, and shape threshold to recognize objects and eliminate noise. The proposed algorithm is tested on a database consisting of 100 images that be divided into five groups. The accuracy of the algorithm can reach more than 85%. There are a few cases of failure due to lighting conditions or some objects being very close to each other. The 3D coordinates of objects are estimated just by solving a linear equation system. The experiment results show that the estimation method based on vision has an average error of 3.47 mm. Using a simple algorithm minimizes the cycle time. The system needs a total time of 2.2 to 2.4s to achieve the sorting of one object. This is the time it takes for the robot to move.

In future works, the robot arm of the system will be updated. The stepper motors are replaced by servo motors for higher movement speed. The number of degrees of freedom of the robot is also increased to be able to perform more complex tasks.

ACKNOWLEDGMENT

We acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

REFERENCES

[1] Wang Z., Li H., and Zhang X.:Construction waste recycling robot for nails and screws: Computer vision technology and neural network approach, Automation in Construction, Vol. 97, No. 8, pp. 220-228, 2019.

[2] Chen, S., Li, Y., Kwok, N.: Active vision in robotic systems: a survey of recent developments, Int.J.Rob.Res, Vol. 30, Vol. 11, pp. 1343–1377, 2011

[3] Zhang, Y., Yin, X., Zou, X. and Zhao, J.:On-line sorting maturity of cherry tomato by machine vision, IFIP AICT Vol. 29, No. 5, pp. 2223–2229, 2009

[4] Omid, O.A., Parviz, M. and Asaad M.:Online tomato sorting based on shape, maturity, size, and surface defects using machine vision, Turkish Journal of Agriculture and Forestry, Vol. 37, pp. 62-68, 2012.

[5] Sofu, M.M., Er, O., Kayacan, M.C. and Cetisli, B.: Design of an automatic apple sorting system using machine vision, Computers and Electronics in Agriculture, Vol. 127, pp. 395-405, 2016.

[6] Rafael, R. et.al: A smart system for the automatic evaluation of green olives visual quality in the field, Computers and Electronics in Agriculture, Vol.179, 2020

[7] Chen, S.W., Skandan, S.S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C.J., and Kumar, V. Counting apples and oranges with deep learning: a data driven approach, IEEE Robotics and Automation Letters, Vol. 2, No. 2, pp. 781-788, 2017

[8] Fiaschi, L., Nair, R., Koethe, U. and Hamprecht, F.A.:Learning to count with regression forest and structured labels, 21st International Conference on Pattern Recognition, pp. 2685–2688, 2012.

[9] Gemert, J.C.V., Verschoor C.R., Mettes, P., Epema, K., Lian, P.K. and Wich, S.: Nature conservation drones for automatic localization and counting of animals, European Conference on Computer Vision, pp. 249–259, 2015.

[10] Zhang, L., Li, W., Liu, C., Zhou, X. and Duan, Q.: Automatic fish counting method using image density grading and local regression, Computers and Electronics in Agriculture, Vol. 179, 2020.

[11] Tian, M., Guo, H., Chen, H., Wang, Q., Long, C. and Ma, Y.: Automated pig counting using deep learning, Computers and Electronics in Agriculture, Vol. 163, 2019.

[12] Liping, Z., Hong, Z., Sikandar, A., Baoli, Y. and Chengyang, L.: Crowd counting via Multi-Scale Adversarial Convolutional Neural Networks, Journal of Intelligent Systems, Vol. 30, pp. 180-191, 2020.

[13] Mitzias, D.A. and Mertzios, B.G.: A neural multi-classifier system for object recognition in robotic vision applications, Measurement, Vol. 36, pp. 315–330, 2004.

[14] Abbood, W.T., Abdullah, O.I., Khalid, E.A.:A real-time automated sorting of robotic vision system based on the interactive design approach, International Journal on Interactive Design and Manufacturing (IJIDeM), Vol. 14, pp. 201-209, 2020.

[15] Sangeetha, G.R., Kumar, N., Hari, P.R., Sasikumar, S.: Implementation of a Stereo Vision based system for visual feedback control of Robotic Arm for space manipulations, International Conference on Robotics and Smart Manufacturing, Procedia Computer Science, Vol. 133, pp. 1066–1073, 2018.

[16] Cong V.D., Hanh L.D., Phuong L.H: Real-time Measurement and Prediction of Ball Trajectory for Ping-pong Robot, 2020 5th International Conference on Green Technology and Sustainable Development (GTSD), pp. 9-14 (2020)

[17] Aneesh, A., Dileep, T.N., Kuriakose, J., Varghese, K., Chacko, J.: Object Sorting Robotic Arm based on Colour and Shape Sensing, International Journal for Scientific Research and Development, Vol. 4, No. 1, pp. 784-787, 2016.

[18] Gornea, D., Popescu, D., Stamatescu, G., Fratila, R. : Monocamera robotic system for tracking moving objects, 2014 9th IEEE Conference on Industrial Electronics and Applications, pp. 1820-1825, 2014.

[19] Šuligoj, F., Šekoranja, B., Švaco, M., Jerbić, B.: Object tracking with a multiagent robot system and a stereo vision camera, Procedia Engineering, Vol. 69, pp. 968-973, 2014.

[20] Abbood, W.T., Hussein1, H.K., and Abdullah, O.I.: Industrial Tracking Camera and Product Vision Detection System, Journal of Mechanical Engineering Research and Developments, Vol. 42, No. 4, pp.277-280, 2019.

[21] Bonilla, I., Mendoza, M., Gonzalez-Galvan, E. J., Chavez-Olivares, C., Loredo-Flores, A., Reyes, F.: Path-tracking maneuvers with industrial robot manipulators using uncalibrated vision and impedance control, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 42, No. 6, pp. 1716-1729, 2012

[22] Jia, B., Liu, S. and Liu, Y.: Visual trajectory tracking of industrial manipulator with iterative learning control, Industrial Robot: An International Journal, Vol. 42, No. 1, pp. 54-63, 2015.

[23] Cong, V.D.: Industrial robot arm controller based on programmable system-on-chip device, FME Transactions, Vol. 49, No.4, pp. 1025-1034, 2021.

[24] Austin, D.: Generate stepper-motor speed profiles in real time, article in Embedded Systems Programming, January 2005. http://www.embedded.com//showArticle.jhtml?articleID=56800129

[25] Calibration Toolbox for Matlab https://au.mathworks.com/help/vision/ref/cameracalibrator-app.html

[26] Abdullah, O.I., Abbood, W.T. and Hussein, H.K.: Development of automated liquid filling system based on the interactive design approach, FME Transactions, Vol. 48, No.4, pp. 938-945, 2020.

[27] Caruana, L. and Francalanza, E.: Safety 4.0 for Collaborative Robotics in the Factories of the Future, FME Transactions, Vol. 49, No. 4, pp. 842-850, 2021

ДИЗАЈН И РАЗВОЈ СИСТЕМА РОБОТСКИХ РУКУ ЗА КЛАСИФИКАЦИЈУ И СОРТИРАЊЕ ПРИМЕНОМ МАШИНСКОГ ВИДА

В.Д. Конг, Л.Д. Хан, Л.Х. Фуонг, Л.А. Дуј

Главни фокус овог рада је пројектовање и развој система од две роботске руке за класификацију и сортирање објеката на основу облика и величине помоћу машинског вида. Систем користи јефтин хијерархијски контролни систем високих перфор–манси укључујући једног главног и два славе. Сваки славе је роботски контролер заснован на микро–контролеру који прима команде од мастера да самостално контролише руку робота. Мастер је уграђени рачунар који се користи за обраду слике, кинематичке прорачуне и комуникацију.

Предложен је једноставан и ефикасан алгоритам за обраду слике који се може применити у реалном времену, помажући да се скрати време процеса сортирања. Предложени метод користи серију алгоритама укључујући проналажење контура, екстракцију границе, алгоритам центроида и праг облика за препознавање објеката и елиминисање буке. 3Д координате објеката се процењују само решавањем система линеарних једначина. Плани–рано је да покрети роботских зглобова прате трапезоидни профил са фазом убрзања/успоравања, чиме се помаже да се роботи крећу глатко и смањују вибрације.

Експериментална процена открива ефи–касност и тачност роботског видног система у процесу сортирања. Систем се може користити у индустријском процесу за смањење потребног времена за постизање задатка производне линије, што доводи до побољшања перформанси произ–водне линије.