Development of a SCARA Robot Arm for Palletizing Applications Based on Computer Vision

Ho Vinh Nguyen
Industrial Maintenance Training Center
Ho Chi Minh City University of Technology
(HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Vo Duy Cong
Industrial Maintenance Training Center
Ho Chi Minh City University of Technology
(HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

Phan Xuan Trung
Industrial Maintenance Training Center
Ho Chi Minh City University of Technology
(HCMUT), VNU-HCM, Ho Chi Minh City
Vietnam

This paper develops a computer vision system integrated with a SCARA robot arm to pick and place objects. A novel method to calculate the 3D coordinates of the objects from a camera is proposed. This method helps simplify the camera calibration process. It requires no knowledge of camera modeling and mathematical knowledge of coordinate transformations. The least square method will predate the Equation describing the relationship between pixel coordinates and 3D coordinates. An image processing algorithm is presented to detect objects by color or pixel intensity (thresholding method). The pixel coordinates of the objects are then converted to 3D coordinates. The inverse kinematic Equation is applied to find the joint angles of the SCARA robot. A palletizing application is implemented to test the accuracy of the proposed method. The kinematic Equation of the robot arm is presented to convert the 3D position of the objects to the robot joint angles. So, the robot moves exactly to the required positions by providing suitable rotational movements for each robot joint. The experiment results show that the robot can pick and place 27 boxes on the conveyor to the pallet with an average time of 2.8s per box. The positions of the boxes were determined with an average error of 0.5112mm and 0.6838mm in the X and Y directions, respectively.

Keywords: *SCARA robot, image processing, palletizing, computer vision, production line.*

1. INTRODUCTION

In recent years, industrial robot arms have been widely used in many applications in the industry. Using the robot arm in an automatic production line has brought more benefits, such as increasing productivity and quality of the product, reducing the number of defective products, and reducing the waste of raw materials. Especially the integration of computer vision in the robot controller has helped increase the ability of robots and helped perform other tasks that required interaction with the environment. In the robotic vision system, the information in the surrounding environment is perceived by cameras [1-5]. A computer processes the vision data to provide useful information to the robot controller. As a result, the robot will complete the requests more accurately and efficiently [6-10].

In a grasping task in a production line, the vision system will detect objects in the image and estimate the 3D position and orientation of objects so the robot can move exactly to the position of objects [11-13]. The performance of the grasping task depends on the accuracy and speed of the detection and estimation process. There have been many studies to develop algorithms for detecting objects in the image and calculating the 3D pose. Detecting an object in the image is a challenging task. Some challenges in a detecting task are familiar and complex stacking scenes, multi-objects, object-cluttered scenes, occlusions, and texture less objects, etc. [14-18]. The objects can be extracted by using their features such as color, intensity, size, shape, etc. These features are used in simple cases where the interested objects are very different compared to the background. In complex scenarios, machine learning or deep learning methods are applied [19-22].

Accurate estimating of the 3D pose of the objects is required for the robot grasping tasks. In most pick-and-place applications in a conventional industrial pro–duction line, the 3D model of the interested objects must be known in advance. So, based on analyzing the shape or extracting the features of the object, the algorithms form equations to calculate the 3D coordinates for industrial robot arms to perform pick and place objects [23-25]. However, these methods are sensitive to illumination change and noise from the surrounding environment. If the 3D model of the objects is not known prior or there are many objects in the workspace to grasp, these approaches are also not applied. The problem of 6D pose estimation is mainly addressed by using depth data or RGB-D images [26-28]. These approaches can achieve high accuracy with a high-quality depth map. However, the depth maps produced from a low-cost depth camera or from a binocular vision system commonly have low accuracy. In recent years, using RGB images from different viewpoints has received more attention. If only using a single RGB image for pose estimation, the results have low accuracy due to the scale, depth, and perspective ambiguities.

Received: July 2023, Accepted: September 2023
Correspondence to: Vo Duy Cong
Industrial Maintenance Training Center
Ho Chi Minh City University of Technology, Vietnam
E-mail: congvd@hcmut.edu.vn
doi: 10.5937/fme2304541N

Combining multiple RGB images from different viewpoints helps enhance the performance of the estimation. With the machine learning paradigm becoming popular, more and more research has been focused on applying the deep learning technique to deal with automatic object-grasping tasks.

Although the deep learning method can deal with the problem of pose estimation well, it requires the training process before being applied in the real application. This is time-consuming and costly. A large amount of data needs to be collected, and a powerful computer is required to train the neural networks.

In this paper, we propose a simple method to calculate the 3D position of the objects on the conveyor. This method helps simplify the camera calibration process. It requires no knowledge of camera modeling and mathematical knowledge of coordinate transformations. The “transform equation” describing the relationship between pixel coordinates and 3D coordinates will be predefined by the least square method. Then, when an object is detected in the image, the pixel centroid is extracted and converted to the 3D space by using the “transform equation”. The proposed method is applied in a palletizing task to grasp boxes moving on the conveyor and place them on a pallet. A 4-DOF SCARA robot is used to pick and place boxes. The kinematics Equation of the robot is presented for workspace analysis and finding the joint angles. The objects moved on a conveyor are detected by color or their intensity using a simple image processing algorithm.

The reminders of this paper are as follows: The kinematics Equation of the SCARA robot arm is presented in Section 2. The forward kinematics is used to analyze the workspace. Section 3 presents the computer vision system. The image processing algorithm is developed for detecting objects, and a novel method is proposed to calculate the 3D position of the objects. A palletizing task is conducted in the Experiment Section. The robot picks the boxes on a conveyor and stacks them on a pallet. Finally, Section 5 is the Conclusion.

2. KINEMATIC EQUATION OF SCARA ROBOT

This section presents the forward kinematic and inverse kinematic problems of the 4-DOF SCARA robot. If we want to calculate the position and orientation of the end-effector from the joint angles, we need the forward kinematic equations. In contrast, the inverse kinematics problem is used to determine the joint angles if we know the position and orientation of the end-effector. The robot workspace can also be analyzed from the kinematic equations. Figure 1 shows the kinetic diagram of the robot arm. The robot consists of a translation motion and three rotation motions. The motion in the Z direction is an independent degree of freedom. So, we only consider the relationship of the rotation angles with the X and Y coordinates. In Figure 1, the lengths of links are l_1, l_2 and l_3, respectively. The joint angles are θ_1, θ_2 and θ_3. The pose of the end-effector is $p = (x_p, y_p, \theta_p)$. From the geometrical relationship, we have:

$$x_p = d + l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) + l_3\cos(\theta_1+\theta_2+\theta_3) \quad (1a)$$

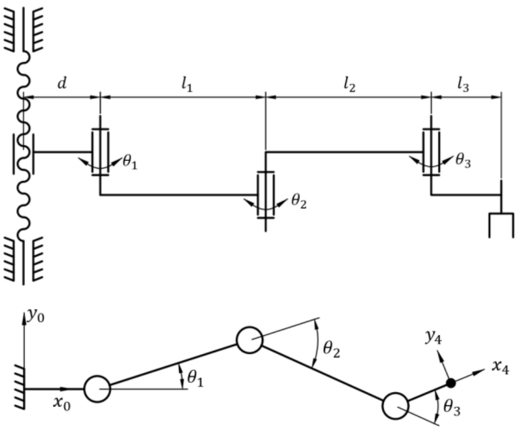


Figure 1. Schematic of the 4-DOF SCARA robot arm

$$y_p = l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) + l_3\sin(\theta_1+\theta_2+\theta_3) \quad (1b)$$

$$\theta_p = \theta_1 + \theta_2 + \theta_3 \quad (1c)$$

Equation (1) is the forward kinematic Equation of the SCARA robot arm. Once there is a specific length of links, for each value of the joint angles, the pose of the end-effector is the only one.

To obtain the inverse kinematic Equation, we solve Equation (1) to find the joint angles θ_1, θ_2, and θ_3. Substitute angle θ_p from Equation (1c) into Equation (1a) and (1b):

$$l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) = x_p - d - l_3\cos\theta_p = a \quad (2a)$$

$$l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) = y_p - l_3\sin\theta_p = b \quad (2b)$$

Square both sides of Equation (2) and add together to get:

$$l_1^2 + l_2^2 + 2l_1l_2\cos\theta_2 = a^2 + b^2 \quad (3)$$

From equation (2), the angle θ_2 can be determined by:

$$\theta_2 = \pm\arccos\left(\frac{a^2+b^2-l_1^2-l_2^2}{2l_1l_2}\right) \quad (4)$$

After having the angle θ_2, expand equation (2a) obtain:

$$(l_1 + l_2\cos\theta_2)\cos\theta_1 - l_2\sin\theta_2\sin\theta_1 = a \quad (5)$$

Denote:

$$r = \sqrt{(l_1 + l_2\cos\theta_2)^2 + (l_2\cos\theta_2)^2}$$
$$\cos\varphi = \frac{l_1 + l_2\cos\theta_2}{r} \quad (6)$$
$$\sin\varphi = -\frac{l_2\sin\theta_2}{r}$$

Equation (5) is converted to:

$$r\cdot\cos(\theta_1 - \varphi) = a \quad (7)$$

So, the value of the angle θ_1 is:

$$\theta_1 = \pm \arccos\left(\frac{a}{r}\right) + \varphi \tag{8}$$

Finally, the value of the angle θ_3 can be calculated from Equation (1c):

$$\theta_3 = \theta_p - \theta_1 - \theta_2 \tag{9}$$

To analyze the workspace of the robot, from equ–ation (1), we have:

$$\left(x_p - d\right)^2 + y_p^2 = l_1^2 + l_2^2 + l_3^2 + 2l_1 l_2 \cos\theta_2 \tag{10}$$

The right-hand side of Equation (10) reaches its ma–ximum value when the cosine of the angles is one, so:

$$\left(x_p - d\right)^2 + y_p^2 \le \left(l_1 + l_2 + l_3\right)^2 \tag{11}$$

So, the maximum reaching distance of the robot is $l_1 + l_2 + l_3$.

In the case the joint angles are limited by the physical constraints, the workspace of the robot is also limited.

To find the exact workspace, we use MATLAB to draw the workspace. Each joint angle is divided into 36 values. Three joints will create 36^3 = 46656 points of the end-effector in the workspace.

A program is written in MATLAB to calculate the 3D position of the end-effector from the joint angles by using Equation (1).

All 3D points are drawn on a graph to show the robot's workspace. With the robot's parameters in Table 1, the result is shown in Figure 2.

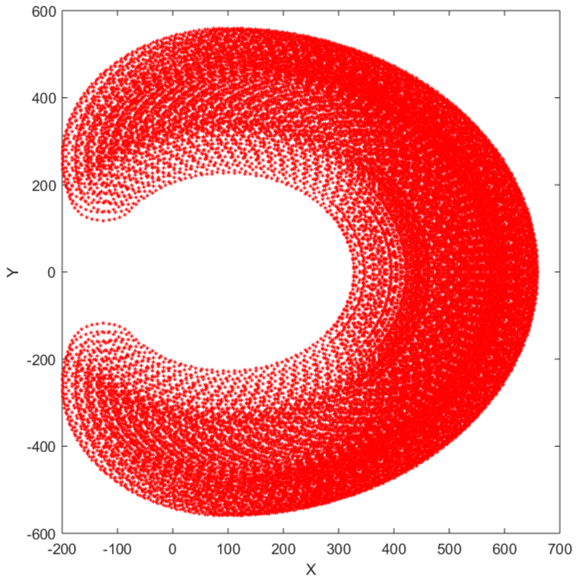


Figure 2. The robot’s workspace

Table 1. The robot parameters

Parameter	Value
Distance d	100mm
The first link’s length	260mm
The second link’s length	240mm
The third link’s length	60mm
The first joint angle	-90^0 to 90^0
The second joint angle	-110^0 to 110^0
The third joint angle	-120^0 to 120^0

3. COMPUTER VISION SYSTEM

The computer vision system helps the robot detect objects and calculate the position of objects. This paper presents an image processing algorithm to detect objects according to their color or apply a thresholding method. Then, a simple method is proposed to calculate the 3D coordinates of the objects. The 3D coordinate is transformed to the robot joint angles using inverse kinematics, and the robot moves to the objects.

3.1 Image processing algorithm

There are many techniques that can be applied to detect an object. It depends on the outstanding features of the object compared to the background or other objects. In this section, we present two methods to detect objects by using the color feature and the thresholding methods.

In the color method, objects are assumed to have a different color compared to the background color. The RGB image of the objects is converted to the HSV image for color classification. The HSV color space is commonly used for detecting an object by color. An image in the HSV color space is represented by three channels: Hue, Saturation, and Value. Each color in the HSV space corresponds to a specific range of H, S, and V. So, to detect an object with a specific color, we apply a threshold for each channel as follows:

$$\begin{aligned} & dst(I) = small(I)_H \le hsv(I) \le high(I)_H \wedge \\ & small(I)_S \le hsv(I) \le high(I)_S \wedge \\ & small(I)_V \le hsv(I) \le high(I)_V \end{aligned} \tag{12}$$

where $hsv(I)$ is the source image in the HSV space, $dst(I)$ is the destination image, $[small(I)_\mathrm{H}, high(I)_\mathrm{H}]$ is the threshold of Hue, $[small(I)_\mathrm{S}, high(I)_\mathrm{S}]$ is the threshold of Saturation, $[small(I)_\mathrm{V}, high(I)_\mathrm{V}]$is the threshold of value.

To transform from an RGB space to an HSV space, we use the following Equation:

$$V = \max\left\{R, G, B\right\} / 255 \tag{13}$$

$$S = \begin{cases} 1 - \dfrac{\min\left\{R, G, B\right\}}{\max\left\{R, G, B\right\}} & if \quad M > 0 \\ 0 & if \quad M = 0 \end{cases} \tag{14}$$

If $G \ge B$:

$$H = \cos^{-1} \frac{R - \dfrac{B+G}{2}}{\sqrt{R^2 + G^2 + B^2 - RG - GB - BR}} \tag{16}$$

If $G \ge B$:

$$H = 360 - \cos^{-1} \frac{R - \dfrac{B+G}{2}}{\sqrt{R^2 + G^2 + B^2 - RG - GB - BR}} \tag{17}$$

In the thresholding method, the interested objects are assumed to have different pixel intensities than other objects. So, when applying a threshold on the image, the objects are distinguished from the background. The main issue is how to determine an exact threshold. There are many techniques to determine the thresholds. Among them, the Otsu method is the most popular method due to its high responsiveness to environmental changes. In the Otsu method, the threshold T is determined by minimizing the within-class variance $\sigma_w^2(t)$:

$$\begin{aligned} &\sigma_w^2 = w_1(t)\sigma_1^2(t) + w_2\sigma_2^2(t) \\ &T = \min_t \sigma_w^2 \end{aligned} \tag{18}$$

Or maximize the between-class variance $\sigma_b^2(t)$:

$$\begin{aligned} &\sigma_b^2 = w_1(t)w_2(t)\left(\mu_1(t) - \mu_2(t)\right)^2 \\ &T = \max_t \sigma_b^2 \end{aligned} \tag{19}$$

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

After applying the HSV threshold or Otsu threshold, the objects are extracted from the background. However, the image still contains some small noisy objects or some small holes in the objects' region. So, the morphological operators are applied to deal with these issues. The closing operator is applied to fill the hole in the objects and to delete the small objects, the opening operator is applied. A dilation followed by an erosion establishes the opening operator. An erosion followed by a dilation establishes the closing operator. In the case of small objects that are not completely deleted, an area threshold is applied to exclude them. In the algorithm, we do not need to process objects with an area smaller than a predetermined threshold.

3.2 Calculating the 3D position of the objects

To calculate the 3D position of objects from their pixel coordinates, it needs to use the intrinsic and extrinsic parameters obtained from the calibration process. In addition, the geometric model of objects is also required. When projected into two-dimensional image space, three-dimensional information has lost information about depth. Therefore, one more constraint is required to reconstruct 3D coordinates from 2D image coordinates. In this paper, we propose a simple method to calculate the 3D position of objects from 2D image points. The objects are placed on the conveyor, so the Z coordinate of the objects is assumed to be known in advance. We use quadratic equations to represent the relationship between 3D coordinates in the X and Y axis with pixel coordinates as follows:

$$X_p = a_1u^2 + a_2v^2 + a_3uv + a_4u + a_5v + a_6 \tag{20a}$$

$$Y_p = b_1u^2 + b_2v^2 + b_3uv + b_4u + b_5v + b_6 \tag{20b}$$

where a_i, b_i are constant coefficients that need to be determined. To find the coefficients a_i, b_i, we collect n points (X_{pi}, Y_{pi}, Z_{pi}) with the pixel coordinates (u_i, v_i). The Equation (20a) written for n points is:

$$\begin{cases} X_{p1} = a_1u_1^2 + a_1v_1^2 + a_3u_1v_1 + a_4u_1 + a_5v_1 + a_6 \\ X_{p2} = a_1u_2^2 + a_2v_2^2 + a_3u_2v_2 + a_4u_2 + a_5v_2 + a_6 \\ \cdots \\ \cdots \\ X_{pn} = a_1u_n^2 + a_2v_n^2 + a_3u_nv_n + a_4u_n + a_5v_n + a_6 \end{cases} \tag{21}$$

Rewritten in the matrix form:

$$\begin{bmatrix} X_{p1} \\ X_{p2} \\ \cdots \\ \cdots \\ X_{pn} \end{bmatrix} = \begin{bmatrix} u_1^2v_1^2u_1v_1u_1v_1 & 1 \\ u_2^2v_2^2u_2v_2u_2v_2 & 1 \\ \cdots & \\ \cdots & \\ u_n^2v_n^2u_nv_nu_nv_n & 1 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} \tag{22}$$

Equation (21) has six unknown values, so it needs at least six image points to accomplish six equations. However, the extracted pixels are distorted due to noise, so using only six will give inaccurate results. For accurate and general results, we must extract more than six points, and Equation (22) is solved by using the least squares method (LSM). Denote:

$$\begin{aligned} &A = \left[a_1a_2a_3a_4a_5a_6\right]^T \\ &M = \begin{bmatrix} u_1^2v_1^2u_1v_1u_1v_1 & 1 \\ u_2^2v_2^2u_2v_2u_2v_2 & 1 \\ \cdots & \\ \cdots & \\ u_n^2v_n^2u_nv_nu_nv_n & 1 \end{bmatrix}, X = \begin{bmatrix} X_{p1} \\ X_{p1} \\ \cdots \\ \cdots \\ X_{pn} \end{bmatrix} \end{aligned} \tag{23}$$

Equation (21) becomes:

$$X = MA \tag{24}$$

According to the least square method, the solution of Equation (24) is:

$$A = \left(M^TM\right)^{-1}M^TY \tag{25}$$

Similarly, the coefficients b_i are determined by:

$$B = \left(M^TM\right)^{-1}M^TY \tag{26}$$

where Y is the vector of the Y-coordinates:

$$Y = \begin{bmatrix} Y_{p1} \\ Y_{p2} \\ \cdots \\ \cdots \\ Y_{pn} \end{bmatrix} \tag{27}$$

3.3 The novelty of the proposed methodology

Calibration is an important step in the computer vision field. Camera parameters need to be specified to cal–culate the 3D coordinates of the objects. This requires a certain amount of camera knowledge (camera model, intrinsic and extrinsic parameters, camera distortion, etc.), knowledge of geometry, methods to convert coordinate systems, matrices, vectors, etc. Therefore, calibration is a time-consuming and labor-intensive job. The normal calibration method will use a checkerboard. Images of the checkerboard in different positions and poses are collected, and software or program is used to process images and calibrate the camera. The calibration process will return the internal and external parameters of the camera. The user will use these parameters together with the camera's model to calculate the object's 3D coordinates. In this paper, to simplify the process of calibrating and estimating 3D coordinates, the "transform equation" describing the relationship between pixel coordinates and 3D coordinates will be predefined by the least square method. Then, when an object is detected in the image, the pixel centroid is extracted and converted to the 3D space by using the "transform equation". The proposed method simply uses an image of the checkerboard and extracts the 2D coordinates of the checkers. Equations describing the relationship between the 2D and 3D coordinates are easy to calculate by the least squares method without using any other program or software.

4. RESULTS AND DISCUSSION

3.4 Experiment setup

Figure 3 shows the experimental setup of a palletizing system. The system consists of a SCARA robot arm to pick up the paper boxes on the conveyor to stack them on a pallet. A camera is used to detect boxes when they move to the end of the conveyor. The size of the pallet is 150 mm x 180 mm. The pallet will contain twenty-seven boxes stacked on three floors. Each floor has nine boxes sorted into three columns and three rows. The size of each box is 50x60x10mm.

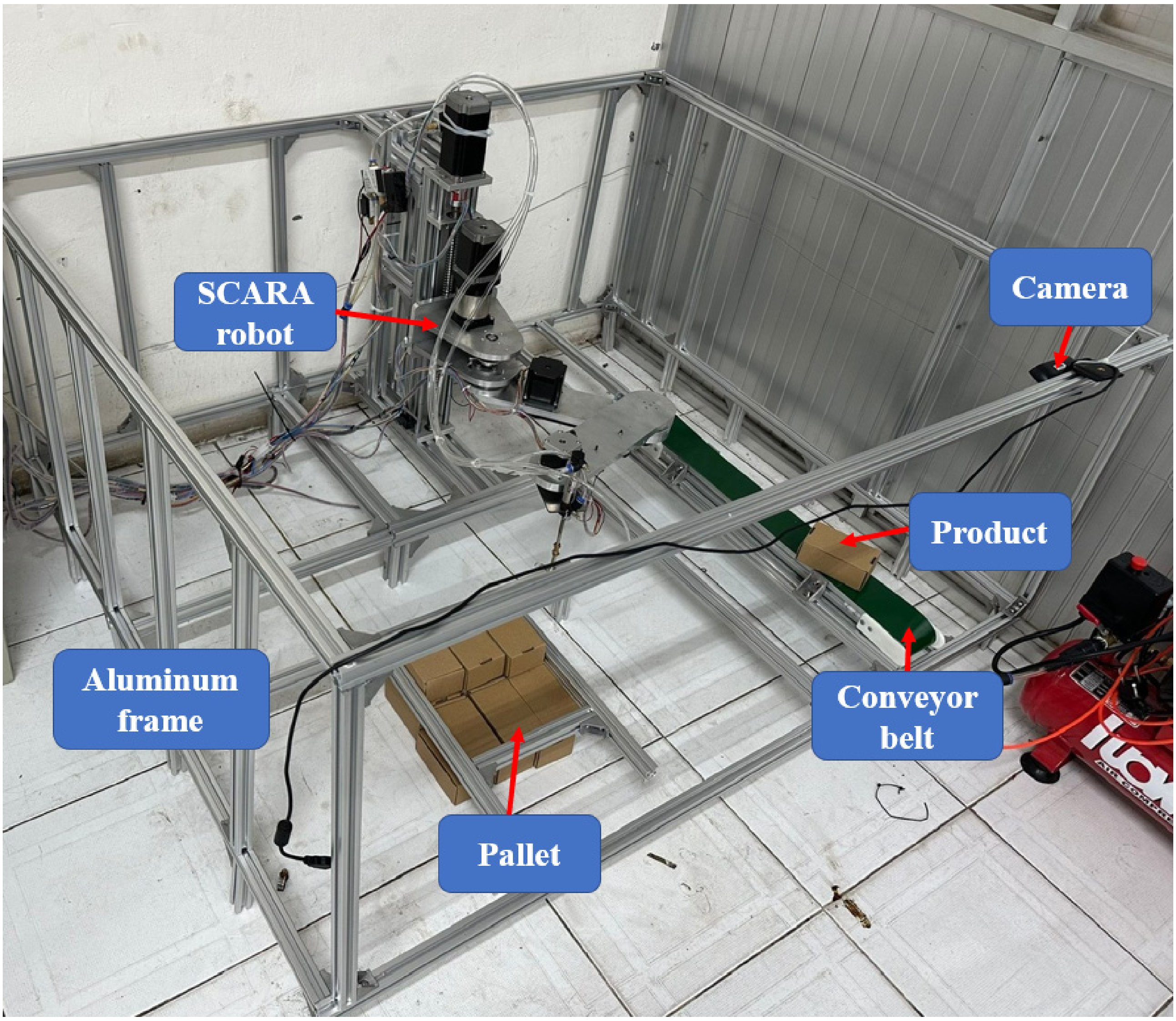


Figure 3. The experiment robot system

Stepper motors drive the SCARA robot arm. The stepper motors are controlled with a micro-step of 6400 pulse/revolution. The maximum holding torque of the stepper motor is 2Nm. They can rotate at a maximum speed of 360^0/s. The robot arm is connected to a screw mechanism to translate in the Z-direction. At the robot's end-effector is mounted a vacuum suction cup to grasp the boxes. The robot arm is controlled by PLC Delta DVP28SV11T2. This PLC has four outputs that can create high-frequency pulses for controlling stepper motors or servo motors.

The camera is fixed above the conveyor to capture images of the boxes. It is a certain distance from the conveyor belt to ensure that it can capture the complete image of the boxes. The obtained images have a resolution of 640x480 pixels. The camera is connected to a computer by a USB port. The image processing algorithm is written in Python programming language with the support of the OpenCV library.

3.5 Results

According to the required task, the robot arm must move to different positions to grasp boxes on the pallet. Each position of the boxes corresponds to different configurations of the robot. We need to calculate the robot's joint angles for each configuration by using the inverse kinematic Equation presented in Section 2. Table 2 shows the joint angles of the robot with nine positions of the boxes on the pallet.

Table 2. The robot joint angles

Cell	X, Y coordinates	Joint angle (θ_1, θ_2, θ_3)
1	(350, -200)	(6.64,-113.167, 106.52)
2	(350, -140)	(21.25, -123.84, 102.59)
3	(350, -80)	(37.84, -131.50, 93.66)
4	(400, -200)	(8.35, -90.51, 82.16)
5	(400, -140)	(21.47, -99.92, 78.45)
6	(400, -80)	(34.60, -106.14, 71.54)
7	(450, -200)	(0.38, -57.58, 57.20)
8	(450, -140)	(12.77, -68.12, 55.36)
9	(450, -80)	(23.93, -74.52, 50.60)

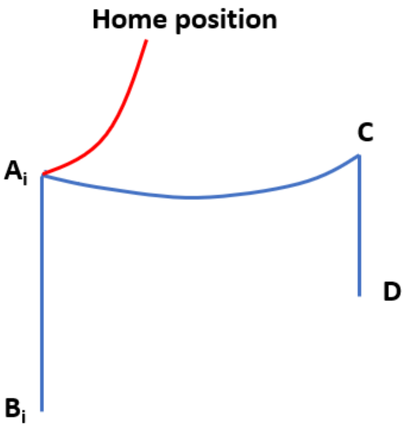


Figure 4. The trajectory of the robot

In a pick-and-place operation, the motion of the robot consists of several phases, as shown in Figure 4. First, the robot moves from the "Home position" to the approach position A_i of the box on the conveyor. Then, the robot moves in a vertical line to the grasping position, Bi, which is determined by the computer vision system. At this position, the box is attached to the robot's gripper. The box is lifted upwards in a straight line to position A_i before moving to point C_i along a curve. Finally, the box is moved to the point D_i on the pallet in a straight line and is dropped on the pallet. To grasp another box, the robot will move from position D_i to the box position B_{i+i}; the robot will move backward from point D_i to point C_i, from point C_i to point A_{i+1}, and from A_{i+1} to B_{i+1}. The process is repeated until the robot has completed grasping 27 boxes and stacks them on the pallet. The points A_i and B_i are determined by the computer vision system, and the points C_i, D_i will change depending on the position of the boxes on the pallet. Moving according to the path in Figure 4 helps the robot avoid colliding with other objects.

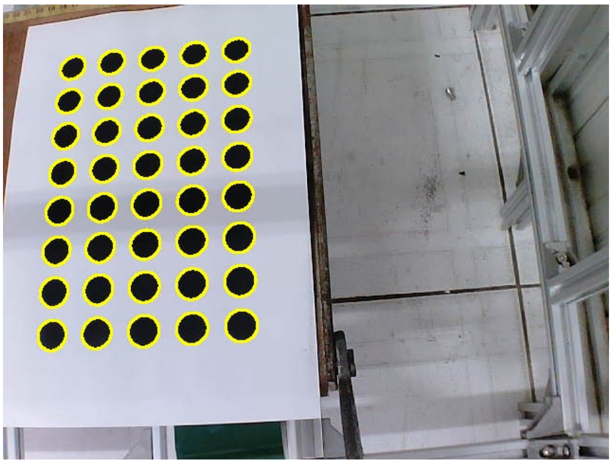


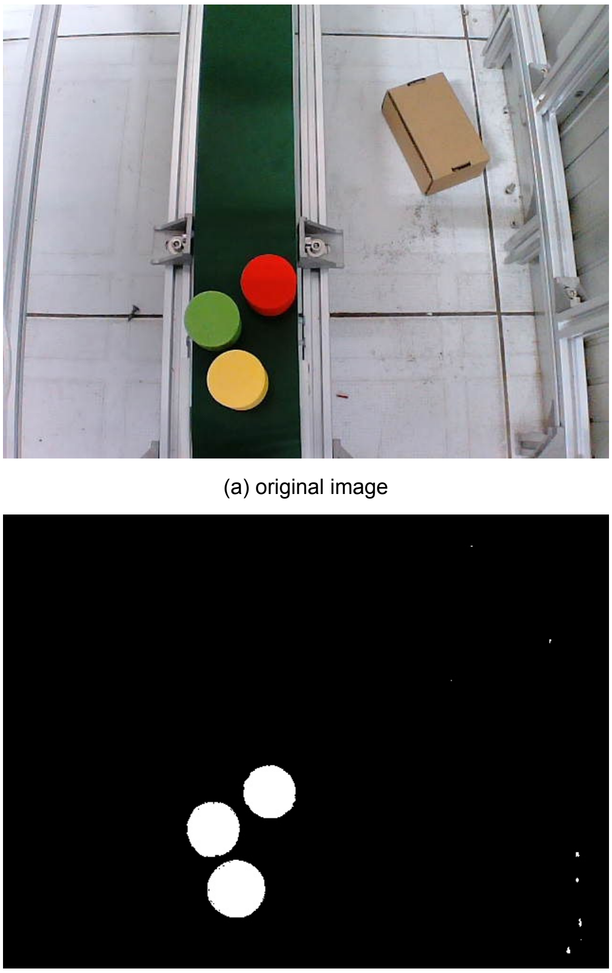
Figure 5. Calibration plate

To calculate the 3D position of the boxes on the conveyor, we must determine the coefficients in Equation (20). A calibration plate with 40 circles is used. The plate is placed on the conveyor; the camera takes the image of the plate. The circles are detected, and their centroids are extracted, as shown in Figure 5. Using Equation (25) and Equation (26), the coefficients are determined as follows:

$$A=\begin{bmatrix}1.0969\times10^{-4}\\-2.1504\times10^{-5}\\2.64423\times10^{-5}\\0.8254\\-0.0202\\-119.8452\end{bmatrix}\quad B=\begin{bmatrix}7.5174\times10^{-6}\\-2.8691\times10^{-5}\\-1.2220\times10^{-4}\\3.82020\times10^{-3}\\-0.8711\\389.6859\end{bmatrix}$$

These coefficients are used to calculate the 3D coordinates of the boxes as the conveyor transports them to the camera's field of view. Table 3 shows the performance of the proposed method. The positions of the objects were determined with an average error of 0.5112mm and 0.6838 mm in the X and Y directions, respectively. When objects are far from the image's centroid, the error will be larger than that of objects in the image's centroid. The maximum error is 2.19mm and 1.82mm in the X and Y directions, respectively.

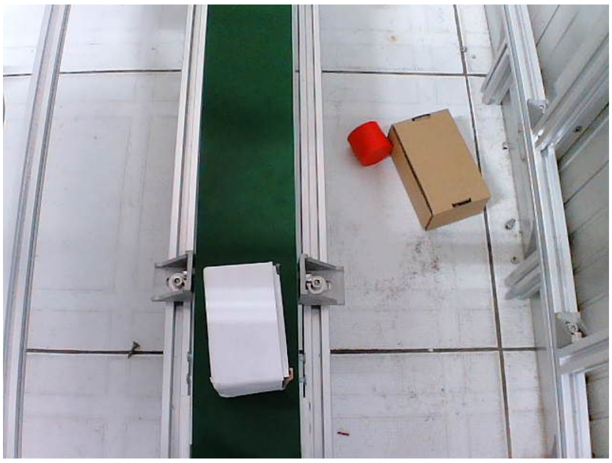
Figure 6 shows the results of detecting the objects by applying the HSV threshold. The thresholds of Hue, Saturation, and Value are determined by changing them until we get a full shape of the object in the binary image. If we want to detect many objects with different colors, we can apply different thresholds to each color in turn. In Figure 6, three objects with three colors (red, yellow, and green) are detected. Figure 7 shows the result of applying the Otsu threshold. The resulting image also has a noisy object due to the lighting conditions. However, this is only a small object which can be eliminated using an area threshold.



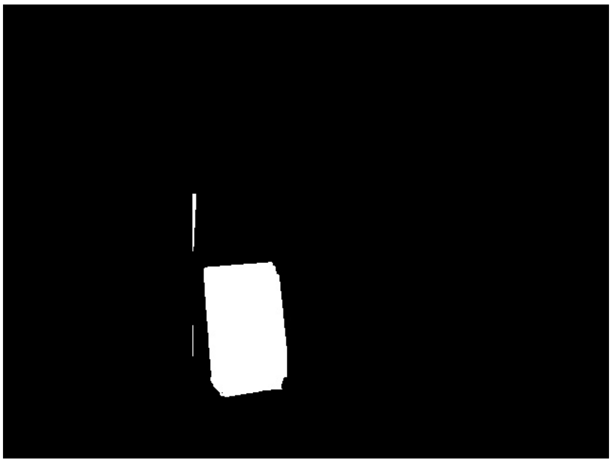
(a) original image

(b) resulting image

Figure 6. The HSV threshold method result



(a) original image



(b) resulting image

Figure 7. The Otsu threshold method result

After detecting the object and calculating the 3D coordinates using the abovementioned methods, the object's position is sent to the robot. The robot performs the process of picking and dropping the object from the conveyor to the pallet according to the trajectory, as shown in Figure 5. The whole process of picking up the product takes about 2.8 seconds, including image processing time.

Table 3. The performance of the estimation method

	X coordinate	Y coordinate
Average error	2.19mm	1.8203 mm
Maximum error	0.5112mm	0.6838mm
Standard Deviation	0.3923mm	0.4324mm

CONCLUSION

This study has developed a SCARA robot system for a palletizing application based on computer vision. The kinematic problem of the robot 4-DOFSCARA arm is analyzed to determine the robot's workspace. As a result, it is possible to place pallets within the robot's operating area. The inverse kinematics problem helps to calculate the value of the robot's joint angles from the 3D position of the gripper.

To ensure the flexibility of the robot system, a com–puter vision system is integrated to determine the object's position on the conveyor. Converting from pixel coordinates to 3D coordinates requires the calibration process to know the camera parameters and the frame relationship between the camera and the robot. This requires camera modeling and mathematical knowledge. In this paper, we use quadratic equations to represent the relationship between 3D coordinates in the X and Y axis with pixel coordinates. This is a simple method with low computational cost and does not require any knowledge of camera modeling and geometric relati–onships.

An experimental robot system was built to verify the proposed algorithm. The robot grasps twenty-seven boxes and stacks them on the pallet. Experimental results show that the positions of the boxes are determined correctly. The system takes about 2.8s to finish picking up a box. The positions of the objects were determined with an average error of 0.5112mm and 0.6838mm in the X and Y directions, respectively.

The maximum error is 2.19mm and 1.82mm in the X and Y directions, respectively

.In future works, the robot system will be applied in a real production line to evaluate more detail on the effectiveness. The accuracy and processing time are also improved to meet the requirements of the real applications. The camera with higher resolution and speed will be replaced; a more powerful computer will be used to increase the processing time.

ACKNOWLEDGMENT

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

[1] Chen, Y.L., Cai, Y.R., Cheng, M.Y.: Vision-Based Robotic Object Grasping—A Deep Reinforcement Learning Approach, Machines, vol. 11, 275, 2023https://doi.org/10.3390/machines11020275.

[2] Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasps, The International Journal of Robotics Research. Vol. 34, pp. 705-724, 2015. doi:10.1177/0278364914549607

[3] Liu, D., Tao, X., Yuan, L., Du, Y and Cong, M.: Robotic Objects Detection and Grasping in Clutter Based on Cascaded Deep Convolutional Neural Network, IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-10, 2022.

[4] Yu, Q.C., Shang, W.W. and Zhang, C.: Object grab detection based on three-level convolutional neural network, robot, vol. 40, no. 5, pp. 762-768, 2018.

[5] Phuong, L.H., Cong, V.D., Hiep, T.T.: Design a Low-cost Delta Robot Arm for Pick and Place Applications Based on Computer Vision, FME Transactions, vol. 51, pp. 99-108, 2023.

[6] Nikola, S., Sasa, Z., Nikola, V., Zoran, D.: Development of the Programming and Simulation System of 4-axis Robot with Hybrid Kinematic, FME Transactions, vol. 50, pp. 403-411, 2022.

[7] Cong, V.D.: Visual servoing control of 4-DOF palletizing robotic arm for vision-based sorting robot system, Int J Interact Des Manuf (2022). https://doi.org/10.1007/s12008-022-01077-8.

[8] Yuanhao, L., Yu, L., Zhiqiang, M., Panfeng, H.: A Novel Generative Convolutional Neural Network for Robot Grasp Detection on Gaussian Guidance, IEEE Transactions on Instrumentation and Measurement, vol.71, pp.1-10, 2022.

[9] Hongkun, T., Kechen, S., Song, L., Shuai, M., Yunhui, Y.: Lightweight Pixel-Wise Generative Robot Grasping Detection Based on RGB-D Dense Fusion, IEEE Transactions on Instrumentation and Measurement, vol.71, pp.1-12, 2022.

[10] Zhang, H., Peeters, J., Demeester, E., Kellens, K.: A CNN-Based Grasp Planning Method for Random Picking of Unknown Objects with a Vacuum Gripper, J. Intell. Robot. Syst. 2021, 103, 1–19.

[11] Hu, C., Guang, C., Zhijun, L., Yingbai, H., Alois, K.:NeuroGrasp: Multimodal Neural Network With Euler Region Regression for Neuromorphic Vision-Based Grasp Pose Estimation, IEEE Transactions on Instrumentation and Measurement, vol.71, pp.1-11, 2022.

[12] Qide, W., Daxin, L., Zhenyu, L., Jiatong, X., Hui, L., Jianrong, T.: A Geometry-Enhanced 6D Pose Estimation Network With Incomplete Shape Recovery for Industrial Parts, IEEE Transactions on Instrumentation and Measurement, vol.72, pp.1-11, 2023.

[13] Zichen, L., Hu, C., Chu, Y., Zikai, Z., Guang, C.: Global-local Feature Aggregation for Event-based Object Detection on EventKITTI, 2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp.1-7, 2022.

[14] Peigen, L., Guang, C., Zhijun, L., Daniel, C., Zhengfa, L., Ruiqi, Z., Alois, K.:NeuroDFD: Towards Efficient Driver Face Detection with Neuromorphic Vision Sensor, 2022 International Conference on Advanced Robotics and Mechatronics (ICARM), pp.268-273, 2022.

[15] Chu, F.J., Xu, R., Vela, P.A.: Real-world multi-object, multigrasp detection, IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3355–3362, 2018.

[16] Zhang, Y., Xie, L., Li, Y., Li, Y.: A neural learning approach for simultaneous object detection and grasp detection in cluttered scenes, Frontiers in Computational Neuroscience, vol. 17, 2023.

[17] Cheon, J., Baek, S., Paik, S.: Invariance of object detection in untrained deep neural networks, Frontiers in Computational Neuroscience, vol.16, 2022.

[18] Jørgensen, T.B., Jensen, S.H.N., Aanæs, H., Hansen, N.W., Kr¨ uger, N.: An adaptive robotic system for doing pick and place operations with deformable objects. J. Intell. Robot. Syst., vol. 94, no. 1, pp. 81–100, 2019.

[19] Chen, Y.L., Cai, Y.R., Cheng, M.Y.: Vision-Based Robotic Object Grasping—A Deep Reinforcement Learning Approach, Machines, Vol. 11, No. 2, 2023.

[20] Chen, J., Liu, H., Zhang, Y., Zhang, D., Ouyang, H., Chen, X.: A multiscale lightweight and efficient model based on yolov7, Applied to citrus orchard. Plants vol. 11, No. 23, 2022.

[21] Liu, N., Guo, C., Liang, R., Li, D.: Collaborative Viewpoint Adjusting and Grasping via Deep Reinforcement Learning in Clutter Scenes, Machines, vol. 10, vol. 12, 2022.

[22] Cong, V.D.: Extraction and classification of moving objects in robot applications using GMM-based background subtraction and SVMs. J Braz. Soc. Mech. Sci. Eng. Vol. 45, 317, 2023. https://doi.org/10.1007/s40430-023-04234-6

[23] Phuong, L.H, Cong, V.D., Hiep, T.T.: Design a Low-cost Delta Robot Arm for Pick and Place Applications Based on Computer Vision, FME Transactions, Vol. 51, No. 1, pp. 99-108, 2023.

[24] Cong, V.D., Hanh, L.D., Phuong, L.H., Duy, D.A.: Design and Development of Robot Arm System for Classification and Sorting Using Machine Vision, FME Transactions, Vol. 50, No. 1, pp. 181-192, 2022.

[25] Cong, V.D., Phuong, L.H.: Design and development of a delta robot system to classify objects using image processing, International Journal of Electrical and Computer Engineering (IJEC), Vol. 13, No. 3, pp. 2669-2676, 2023.

[26] Yin, R., Wu, H., Li, M., Cheng, Y., Song, Y., Handroos, H.: RGB-D-Based Robotic Grasping in Fusion Application Environments, Applied Sciences, Vol. 12, No. 15, 2022.

[27] Song, Y., Wen, J., Liu, D. et al.: Deep Robotic Grasping Prediction with Hierarchical RGB-D Fusion, Int. J. Control Autom. Syst., Vol. 20, pp. 243–254, 2022.

[28] Shi Y et al.: Symmetry grasp: Symmetry-aware antipodal grasp detection from single-view RGB-D images, IEEE Robotics and Automation Letters Vol. 7, No. 4, pp. 12235–12242.

NOMENCLATURE

x_p	X coordinate of the robot end-effector
y_p	Y coordinate of the robot end-effector
θ_p	orientation of the robot end-effector
θ_1	The first joint angle
θ_2	The second joint angle
θ_3	The third joint angle
$\boldsymbol{S}$	The saturation value
H	The Hue value
V	The value
u, v	The pixel coordinates
σ_w^2	The within-class variance
σ_b^2	The between-class variance

Abbreviations

c	cosine
s	sine
SCARA	Selective compliance assembly robot arm
PLC	Programable Logic Controller
DOF	Degree of Freedom
2D	Two dimensional
3D	Three dimensional
LSM	Least Squares Method
HSV	Hue, Saturation, Value
RGB	Red, Green, Blue

РАЗВОЈ СЦАРА РОБОТСКЕ РУКЕ ЗА АПЛИКАЦИЈЕ ЗА ПАЛЕТИЗАЦИЈУ ЗАСНОВАНЕ НА КОМПЈУТЕРСКОМ ВИДУ

Х.В. Нгујен, В.Д. Конг, Ф.К. Трунг

Овај рад развија систем компјутерског вида интегрисан са СЦАРА роботском руком за бирање и постављање објеката. Предложена је нова метода за израчунавање 3Д координата објеката помоћу камере. Овај метод помаже да се поједностави процес калибрације камере. Не захтева познавање моделирања камере и математичко знање о трансформацијама координата. Метода најмањег квадрата претходиће једначини која описује однос између координата пиксела и 3Д координата. Представљен је алгоритам за обраду слике да детектује објекте по интензитету боје или пиксела (метода прага). Координате пиксела објеката се затим конвертују у 3Д координате. Инверзна кинематичка једначина се примењује за проналажење зглобних углова СЦАРА робота. Примењује се апликација за палетизацију да би се тестирала тачност предложене методе. Представљена је кинематичка једначина руке робота за претварање 3Д положаја објеката у углове зглоба робота. Дакле, робот се помера тачно у тражене позиције обезбеђујући одговарајуће ротационе покрете за сваки зглоб робота. Резултати експеримента показују да робот може покупити и поставити 27 кутија на транспортер до палете са просечним временом од 2,8с по кутији. Положаји кутија су одређени са просечном грешком од 0,5112мм и 0,6838мм у Кс и И смеру, респективно.