An image vision and automatic calibration system for universal robots

Bo-Lin Jian, Chi-Shiuan Tsai, Ying-Che Kuo and Yu-Sying Guo

Abstract
The rapid development of technology is causing the replacement of many traditional manufacture industries by automation. Robotic arms are now commonly used in many sectors. The requirements for robotic arms in different sectors are quite different; but, in general, robotic arms not only save cost and man power, they also improve safety. The aim of this study was an investigation of the integration of image identification with robotic arms. To do this, the Denavit–Hartenberg transformation matrix was used to analyze the mechanical kinematics of the joints of each robotic arm axis. This allowed the spatial relationships between the Cartesian 3D coordinate system and the joint of each axis to be determined and communication between the robotic arm and image identification to be established. A robotic arm prototype platform with automatic image identification and calibration was constructed using a quick and robust method. Several of the variables that exist in real robotic arm applications have been solved in this study: primarily the accuracy error that occurs when repeated gripping of workpieces is done and a movement and placement track is set up. Deviations occur frequently and if image identification can be applied to offset repeated inaccuracy, the quality of finished products will be more consistent. The Universal Robots 5 six-axis robot and cameras, which use a filter, Sobel computation, and Hough transformation computer vision image processing technology, together comprise a system that can automatically identify the workpiece and carry out loading and unloading accurately.

Keywords
Robotic arm, image identification, Hough transformation

Introduction
The purpose of the use of robotic arm is to effectively reproduce the functions of a human arm and to carry out operations under automatic control. These arms are now commonly used on many different industrial production lines. Their primary function is to carry out repetitive work more efficiently and faster than humans. The most obvious difference between a robotic arm and normal automatic machinery is the multiple links and joints. With these functions, the robotic arm can freely move in three-dimensional (3D) space. Most robotic arms have four main parts: the main machine, controller, servo mechanisms, and sensors. The robotic arm used in this paper was a UR5 six-axis robot (Universal Robot 5). The six-axis control involves the kinematics of the Denavit–Hartenberg (D–H) transformation matrix, dynamics, as well as control theory, including the computation of six-axis rotational angles from the work point and pose of the robotic arm and the derivation of spatial work point position and pose from rotational angles to complete spatial track planning, etc.

The Hough transformation was used for image feature detection and identification and is widely used in computer vision, graphic analysis, and digital image processing. Its algorithm process is roughly as follows: when an object needs to be identified, the type can be found from the image and easily derived using a standard Hough transformation.
The algorithm decides the object shape using an edge detection method and by voting in the parameter space. The Hough transformation not only detects lines in an image, but will also identify shapes: circle, oval, triangle, etc. Imperfection of the image, or of edge detection, will cause errors. A noisy image will cause points or pixels to be missed, and the boundary obtained from the edge detector will deviate from the actual boundary. This makes it impossible to intuitively divide the detected edge into an aggregate of a line, circle, or oval. However, the Hough transformation can solve this problem. The voting steps in the Hough transformation algorithm allow graphic parameters to be found in complex parameter space. The computer can determine the shape of the edge from the parameters.

Robotic arm control and method\textsuperscript{14} has been compared with human poses using image detection. However, in this study, the robotic arm did not mimic human movement; it determined and planned the operation autonomously. The robot determines the route to be followed, the deviations needed for calibration, and then completes the task of identification and gripping.\textsuperscript{15,16}

**Methods**

The framework of a robotic arm integration system has two components: hardware and software. The main function of the software is to integrate communications between the hardware and the user end. In this study, a CCD camera was linked to the computer, and a command was sent to the robotic arm for execution. Upon system activation, an image of the work space from the CCD camera is sent to the computer and displayed on the screen after image processing by the Hough transformation. The computer uses the coordinate information to adjust the pose of the robotic arm and complete image identification and pose adjustment to perform the gripping task. The hardware framework consists of acquisition, main control, and the controlled ends. Acquisition is done by the CCD camera, control by the PC, and the robotic arm comprises the controlled end. The hardware system framed network is shown in Figure 1.

**Robotic arm kinematics**

The first step in robotic arm operation is accurate marking of the target object position as well as the relative position of the robotic arm to the target object. In other words, the orientation relationship between them must be determined. After gaining the necessary information about the coordinates, each stage of coordinate transformation can be carried out. Kinematics is used to describe the tracking of an object in motion and involves forward and inverse kinematics.\textsuperscript{17} Forward kinematics determines the robotic arm pose and end point via the rotational angle of each axis joint. Inverse kinematics back calculates the joint angle of each axis from the robotic arm pose and end point position. Forward kinematics is the basis of inverse kinematics and can be used for checking the correctness of inverse kinematics. In industrial applications, a series of end point positions are often given. Inverse kinematics is used to determine the corresponding joint angle of each axis to each position and to control the motors, so that each joint can reach the determined position. This allows the robotic arm to complete each given pose. The kinematic relationship is shown in Figure 2.

![Figure 1. System framework of hardware commands.](image-url)
The D–H forward kinematics robotic arm coordinate system

Forward kinematics involves sending position angle information to the controller for each joint to establish a position. In the D–H coordinate method, appropriate parameters are defined which make originally complex transformation simple and clear. The UR5 robot positions and local reference coordinates of each axis joint\textsuperscript{18–21} are shown in Figure 3.

Standard D–H parameter, \(i\)-robotic arm axis joint, \(\theta\)-joint angle, \(d\)-linkage offset, \(a\)-linkage length, \(z\)-linkage torsion, unit: mm, or rad are shown in Table 1.\textsuperscript{8,18,21,22}

Via the D–H transformation matrix, equations shown in equations (1) to (4)

\[
T(z, \theta_i) = \begin{bmatrix}
\cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\
\sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(1)

\[
T(z, d_i) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & d_i \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(2)

\[
T(x, a_i) = \begin{bmatrix}
1 & 0 & 0 & a_i \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(3)

\[
T(x, a_i) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(a_i) & -\sin(a_i) & 0 \\
0 & \sin(a_i) & \cos(a_i) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(4)

Therefore, the result of each axis via the coordinate system transformation matrix \(i^{-1}T\) can be determined from the above four transformation matrices as shown in equation (5)

\[
i^{-1}T = \begin{bmatrix}
\cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\
\sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & d_i \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & a_i \\
0 & \cos(a_i) & -\sin(a_i) & 0 \\
0 & \sin(a_i) & \cos(a_i) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(5)
It is shown in equation (6) after rearranging equation (5)

\[
\begin{bmatrix}
\cos(\theta_i) & -\sin(\theta_i)\cos(a_i) & \sin(\theta_i)\sin(a_i) & a_i\cos(\theta_i) & \alpha_i \\
\sin(\theta_i) & \cos(\theta_i)\cos(a_i) & -\cos(\theta_i)\sin(a_i) & a_i\sin(\theta_i) & \\
0 & \sin(a_i) & \cos(a_i) & d_i & \\
0 & 0 & 0 & 1 & \\
\end{bmatrix}
\]  

(6)

By combination with the (Table 1) UR robot D–H parameters, we can determine the transformation matrix of each axis joint as shown in equations (7) to (12)

\[
0^0T = \begin{bmatrix}
\cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\
\sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\
0 & 1 & 0 & d_1 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(7)

Table 1. UR robot D–H parameters.

| i | \(\theta_i\) | \(d_i\) | \(a_i\) | \(\alpha_i\) |
|---|---|---|---|---|
| 1 | \(\theta_1\) | 89.159 | 0 | \(\pi/2\) |
| 2 | \(\theta_2\) | 0 | 425 | 0 |
| 3 | \(\theta_3\) | 0 | 392.25 | 0 |
| 4 | \(\theta_4\) | 109.15 | 0 | \(\pi/2\) |
| 5 | \(\theta_5\) | 94.65 | 0 | \(-\pi/2\) |
| 6 | \(\theta_6\) | 82.3 | 0 | 0 |
Finally, the transformation matrix is determined as shown in equation (13)

\[
T = 0^0_T \cdot 1^1_T \cdot 3^2_T \cdot 4^3_T \cdot 6^5_T = \begin{bmatrix}
    n_x & o_x & a_x & p_x \\
    n_y & o_y & a_y & p_y \\
    n_z & o_z & a_z & p_z \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]

After simplification, the parameters are as in equation (14)

\[
\begin{align*}
    s_i &= \sin(\theta_i), \\
    c_i &= \cos(\theta_i), \\
    s_{ij} &= \sin(\theta_i + \theta_j), \\
    c_{ij} &= \cos(\theta_i + \theta_j), \\
    1 \leq i, j \leq 6
\end{align*}
\]

where parameters are shown in equation (15)

\[
\begin{align*}
    n_x &= s_1 s_5 c_6 - c_1 s_23 (c_4 s_6 + s_4 c_5 c_6) - c_1 c_23 (s_4 s_6 - c_4 c_5 c_6) \\
    n_y &= -s_1 s_23 (c_4 s_6 + s_4 c_5 c_6) - s_1 c_23 (s_4 s_6 - c_4 c_5 c_6) - s_5 c_1 c_6 \\
    n_z &= -s_23 (s_4 s_6 - c_4 c_5 c_6) + c_23 (c_4 s_6 + s_4 c_5 c_6) \\
    o_x &= -s_1 s_5 s_6 - c_1 s_23 (c_4 c_6 - s_4 c_5 s_6) - c_1 c_23 (s_4 c_6 + c_4 c_5 s_6) \\
    o_y &= -s_1 s_23 (c_4 c_6 - s_4 c_5 s_6) + c_23 (s_4 s_6 + c_4 c_5 s_6) \\
    o_z &= -s_23 (c_4 c_6 + c_4 c_5 s_6) + c_23 (s_4 c_6 - c_4 c_5 s_6) \\
    a_x &= s_1 c_5 - c_1 c_4 s_5 c_3 + c_1 s_4 s_5 s_23 \\
    a_y &= s_1 s_4 s_5 s_23 - s_1 c_4 s_5 c_3 - c_1 c_5 \\
    a_z &= -s_4 s_5 c_23 - c_4 s_5 s_23 \\
    p_x &= a_2 c_1 c_2 + a_5 c_1 c_3 + d_3 s_1 + d_5 s_1 s_23 + a_6 (s_1 c_5 - c_1 s_23) \\
    p_y &= a_2 s_1 c_2 + d_3 s_1 c_3 - d_4 c_1 + d_5 s_1 s_23 - a_6 (c_1 c_5 + s_1 s_23) \\
    p_z &= a_2 s_2 + a_3 s_23 + d_1 - d_5 c_23 - d_6 s_3 s_23
\end{align*}
\]
When all \( \theta \) are known, from equation (13), we can obtain the UR5 robot rotational matrix \( \mathbf{R} \) as in equation (16) and the final coordinate matrix \( \mathbf{P} \) as in equation (17)

\[
\mathbf{R} = \begin{bmatrix}
  n_x & o_x & a_x \\
  n_y & o_y & a_y \\
  n_z & o_z & a_z \\
\end{bmatrix}
\tag{16}
\]

\[
\mathbf{P} = \begin{bmatrix}
  p_x \\
  p_y \\
  p_z \\
\end{bmatrix}
\tag{17}
\]

**Robotic arm inverse kinematics**

If the pose of the robotic arm end point relative to the machine base coordinate is known, we can back calculate the joint angle \( \theta_i \) of each axis, by inverse kinematics.\(^9\) First, define \( \mathbf{P}_{uv} \) as the vector of the origin coordinate of the joint of the \( v \)th axis to that of the joint of the \( u \)th axis. Also, use the coordinate of the joint of the \( u \)th axis to represent its vector is shown in equation (18)

\[
\mathbf{P}_{uv} = \begin{bmatrix}
  (P_{uv})_x \\
  (P_{uv})_y \\
  (P_{uv})_z \\
\end{bmatrix}
\tag{18}
\]

Since \( \mathbf{P}_0^0 \) and \( ^0 \mathbf{T} \) are known, we can determine \( \mathbf{P}_5^0 \) as shown in equation (19) from Table 1

\[
\mathbf{P}_5^0 = \mathbf{P}_6^0 + ^0 \mathbf{T} \begin{bmatrix}
  0 \\
  0 \\
  -d_6 \\
  1 \\
\end{bmatrix}
\tag{19}
\]

Since \( \mathbf{P}_5^0, \mathbf{P}_5^0, \) and \( ^0 \mathbf{T} \) are known, Table 1 can be used again, as well as the auxiliary angles of \( \phi \) and \( \psi \) from Figure 4, to determine \( \theta_1 \) as shown in equations (20) to (22)\(^{23,24}\)

\[
\phi = \pm \cos^{-1}\left( \frac{d_4}{\sqrt{(P_5^0)_x^2 + (P_5^0)_y^2}} \right)
\tag{20}
\]

\[
\psi = \arctan\left( \frac{(P_5^0)_y}{(P_5^0)_x} \right)
\tag{21}
\]

\[
\theta_1 = \phi + \psi + \frac{\pi}{2}
\tag{22}
\]

Since \( \mathbf{P}_6^0, \mathbf{P}_5^0, ^0 \mathbf{T}, ^1 \mathbf{T}, \) and \( \theta_1 \) are known, we can determine \( \mathbf{P}_6^1 \) as shown in equation (23) from Table 1

\[
\mathbf{P}_6^1 = (^0 \mathbf{T})^{-1} \mathbf{P}_6^0
\tag{23}
\]

From Figure 5, we can determine \( \theta_5 \) as shown in equation (24)

\[
\theta_5 = \pm \cos^{-1}\left( \frac{(P_6^1)_z - d_4}{d_6} \right)
\tag{24}
\]
Figure 4. The planar projection of the fifth axis joint coordinate to the base coordinate.

Figure 5. The planar projection of the sixth axis joint coordinate to the base coordinate.
Since $P_0^1, P_0^2, P_5^0, P_5^1, 0^T, 0^T, \theta_1, \theta_5$ are known, Table 1 is used again to determine $\frac{1}{6}T$ as shown in equation (25)

$$\frac{1}{6}T = (\frac{0}{1}T)^{-1}0^T = \begin{bmatrix}
\cdots \\
\cdots \\
\cdots \\
x & \beta & \gamma \\
0 & 0 & 0 & 1
\end{bmatrix}$$

(25)

where $x, \beta, \gamma$ are shown in equations (26) to (28)

$$x = s_5\epsilon_6$$

(26)

$$\beta = s_5s_6$$

(27)

$$\gamma = \epsilon_5$$

(28)

Therefore, when $\theta_5$ is neither zero nor $\pi$, we can derive equation (29) as

$$\theta_6 = \text{atan2} \left( \frac{-\beta}{x} \right)$$

(29)

When $\theta_6$ is known, $\frac{6}{6}T$ can also be known. When $\frac{6}{6}T$ and $\theta_5$ are known, we can determine $\frac{7}{7}T$, and equations (30) to (32) can be derived

$$\sin \theta_6 = 0$$

(30)

$$\frac{1}{4}T = \frac{6}{6}T (\frac{7}{7}T)^{-1}$$

(31)

$$P_1^4 = P_1^4 + \frac{1}{4}T \begin{bmatrix}
0 \\
-d_4 \\
0 \\
1
\end{bmatrix}$$

(32)

Now, $P_3^4, P_4^4, P_6^0, P_6^1, 0^T, 0^T, \theta_3, \theta_4, \theta_5$ are known. Using Figure 6, we can derive $\theta_3$ and $\theta_2$ as shown in equations (33) to (34), where $\epsilon, \delta, \zeta$ are auxiliary angles

$$\theta_3 = \pm \cos^{-1} \left( \frac{||P_3^4||^2 - (a_2)^2 - (a_3)^2}{a_2a_3} \right)$$

(33)

$$\theta_2 = -\text{atan2} \left( \frac{(P_3^4)^x}{(P_3^4)^x} \right) + \sin^{-1} \left( \frac{a_2 \cos \theta_3}{||P_3^4||} \right)$$

(34)

When $\theta_2$ and $\theta_3$ are known, $\frac{2}{6}T$ and $\frac{3}{6}T$ will be known. Therefore, we can obtain equation (35)

$$\frac{3}{3}T = \frac{1}{2}T\frac{3}{3}T$$

(35)

Last, using equations (36) and (37), we can determine $\theta_4$ as shown in equation (38)

$$\frac{3}{4}T = (\frac{3}{4}T)^{-1}4T = \begin{bmatrix}
\lambda & 0 & 0 \\
\mu & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}$$

(36)
The six-axis robotic arm is driven by six individual motors that provide axial motion at each axis. The arm has six degrees of freedom in 3D space. Theoretically, the end point position of the six-axis robotic arm can reach any position and angle in 3D space. However, it can still get stuck in certain positions because some singularities remain. When it is located at a singularity, inverse kinematics can fluctuate. This poses a potential danger to later tracking planning. When the end point is near a singularity, a minute displacement change can cause a dramatic angle change to some axes. This can generate an approximate infinite angular velocity. If the inverse kinematics of the arm has a solution in the analytic form, then it is possible to find the possible singularities by judging the abnormal condition where the denominator of the algebraic form of the solution is zero. When the value of the matrix determinant of the arm is zero, the robotic arm is located at a singularity. If the second, third, and the fourth axis are in parallel, in an arm with six degrees of freedom, the solution in analytic form that exists in inverse kinematics is given by equation (29), and when \( s_5 = \sin \theta_5 = 0 \), calculation will show the value of the robotic arm matrix determinant to be zero. This means that when \( \sin \theta_5 = 0 \), the robotic arm is positioned at a singularity. Now, \( \theta_6 \) can be any value to fulfill the given constraint. For this, it is necessary that processing of \( \theta_6 \) near the singularity be smooth to avoid fluctuation \( \theta_6 \) and also to guarantee motion accuracy. As mentioned, the singularity can be solved using fuzzy motion planning, constrained quadratic optimization, intermediate desired value approach, and the evolution algorithm.

**Analysis of the kinematic singularity**

The six-axis robotic arm is driven by six individual motors that provide axial motion at each axis. The arm has six degrees of freedom in 3D space. Theoretically, the end point position of the six-axis robotic arm can reach any position and angle in 3D space. However, it can still get stuck in certain positions because some singularities remain. When it is located at a singularity, inverse kinematics can fluctuate. This poses a potential danger to later tracking planning. When the end point is near a singularity, a minute displacement change can cause a dramatic angle change to some axes. This can generate an approximate infinite angular velocity. If the inverse kinematics of the arm has a solution in the analytic form, then it is possible to find the possible singularities by judging the abnormal condition where the denominator of the algebraic form of the solution is zero. When the value of the matrix determinant of the arm is zero, the robotic arm is located at a singularity. If the second, third, and the fourth axis are in parallel, in an arm with six degrees of freedom, the solution in analytic form that exists in inverse kinematics is given by equation (29), and when \( s_5 = \sin \theta_5 = 0 \), calculation will show the value of the robotic arm matrix determinant to be zero. This means that when \( \sin \theta_5 = 0 \), the robotic arm is positioned at a singularity. Now, \( \theta_6 \) can be any value to fulfill the given constraint. For this, it is necessary that processing of \( \theta_6 \) near the singularity be smooth to avoid fluctuation \( \theta_6 \) and also to guarantee motion accuracy. As mentioned, the singularity can be solved using fuzzy motion planning, constrained quadratic optimization, intermediate desired value approach, and the evolution algorithm.

**Hough transformation and mass center calculation**

In this study, the Hough transformation method was used to find the location of the image of the workpiece to be gripped and also to determine the mass center of the workpiece. After processing, this information was transferred to the robotic arm controller to complete the gripping task.

The Hough transformation, also called Hough gradient method, is a special detection method widely used in digital image processing and computer vision analysis. The Hough transformation uses binarization to identify specific lines in the image. The theory is the location of line parameters using the positions of all the points in the image. One-to-many mapping of each point in the image to the parameter space is used to get the possible parameter values. The parameter values generated by all points are accumulated to determine the line parameter that presents most obviously in the parameter space.
The thinking behind the Hough circular transformation\textsuperscript{29–32} is that each non-zero pixel point is likely to be a point on a circle, and an accumulated coordinate plane is generated by voting. The position of the circle is determined by setting up an accumulated weight. On the line coordinate, the circular equations of all points on the same circle are the same. Therefore, when mapping to the 3D space Cartesian coordinate system, also known as the \textit{abr}-coordinate system, they will be the same point. Therefore, by judging the accumulated intersections of each point in the \textit{abr}-coordinate system, the points greater than a certain threshold value are circles. Any straight line that passes and holds a right angle to the circumference will intersect and be perpendicular to a point. The intersection of the points is the center of the circle. The threshold value can be judged from the number of intersections on the circle center, as shown in Figure 7.

For a standard Hough transformation,\textsuperscript{33} three parameters are needed to describe a circle, the circle center, and the radius \((a, b, r)\). Any point \((x, y)\) in the image space can be mapped to 3D space \((a, b, r)\) to form a circular cone as shown in equation (39)

\[
f((x, y), (a, b, r)) = (x - a)^2 + (y - b)^2 - r^2 = 0
\]  

(39)

Look for a possible circle center for the point on each edge and build a 3D circular cone accumulator by using the edge as the circle center. These circles will overlap on the target circle center. Last, find the circle center and the radius from the point with most overlap. The circle can be expressed as in equation (40)

\[
(x - a)^2 + (y - b)^2 = r^2
\]  

(40)

Analyze any pixel point in the graph corresponding to the graph in 3D space. If the radius \(r\) is certain, the pixel point corresponding to the 3D space will be a circle. If the radius \(r\) is not certain, then any shape with the radius \(r\) must be a circle. This circle will form a 3D circular cone as shown in Figure 8.

If a line is drawn along the gradient direction from any point on the circle, then the intersection of all lines may possibly be the circle center. First, calculate the possible circle center. Then, calculate the possible radius with the circle center. The circle that is checked must have a boundary. Detect the edge with the Sobel operator.\textsuperscript{34,35} Technically, the Sobel operator is a discrete difference operator used to find the approximate gradient value of the graphic brightness function. This operator includes two sets of 3 \(\times\) 3 matrices, which are lateral and longitudinal. The Hough circular transformation process has two steps. First, the results and procedures of circle detection are shown in Figures 9 and 10. Then, the procedures and results of radius detection are shown in Figures 11 and 12.

The original color image (Figure 9(a)) processes the original image into a grayscale image (Figure 9(b)). Then, the Sobel operator is used to get the horizontal and vertical gradients (see Figure 9(c) and vertical (Figure 9(d)). Finally, the horizontal and vertical gradient values are combined to find the center point (Figure 9(e)).
Figure 8. Schematics of a circular cone from the Hough transformation.

Figure 9. Detection of the circle: (a) original image; (b) grayscale image; (c) horizontal gradient via Sobel processing; (d) vertical gradient via Sobel processing; (e) Integration of Sobel horizontal and vertical gradients.
Figure 10. The circle detection process.

Figure 11. Circle radius and circle center estimation flowchart.
Figure 10 demonstrates the Detection Circle Process. First, the original color image is converted to grayscale and binary images are obtained from them. The Sobel operator is then used to obtain horizontal and vertical gradient values, which are then mapped to the $abr$ coordinate system accumulator. Finally, all $N(a,b)$ accumulators are compared with the threshold to find the circle in the image.

Figure 11 illustrates the Circle Radius and Circle center Estimation Flowchart. The radius of each gradient value obtained from the gradient image is detected, and the value obtained from detection is saved in the $N(r)$ radius accumulator and compared with the threshold value to obtain the most suitable radius and the center of the circle.

Movement in the positive or negative directions of the gradient is made to carry out detection within the range of the radius, see Figure 12(a). The points recorded after detection are assumed to be the center of circles, see Figure 12(b), and comparisons will result in the strongest being left, see Figure 12(c). Last, the radius using some of the assumed circle centers is calculated to find the optimum circle (Figure 12(d)). Last, Hough transformation is used to obtain the space graph, see Figure 12(e).

After obtaining the optimum circle, we can get the planar transform relationship between the workpiece to be gripped and the current image. The center of this circle can also be defined as the mass center. The mass center is the reference point for control, and its 2D planar coordinates represent the position of the workpiece to be gripped. The robot arm can now be directed to complete the gripping task.

**Figure 12.** Circle radius estimation: (a) detect the radius in the positive and negative gradient directions; (b) assumed circle centers; (c) remove some circle centers; (d) seek the optimum circle center; (e) Hough space.
Communication method-socket

To satisfy the communication requirements\textsuperscript{36,37} for the robotic arm system, and to achieve support for some TCP/IP functions that are not easy to integrate into the prevailing function library (TCP/IP Protocol Suite), we define some system call functions—Socket Interface. The Socket web framework is shown in Figure 13. The TCP/IP Socket protocol used in this study forms a satisfactory link between the PC end and the controlled end of the robotic arm.

The application layer is similar to that used by other network applications such as Telnet, FTP, WWW, POP Email, News, and BBS. The transport layer is responsible for network connection, division, and combination for transmitting messages and for providing control for data traffic between usage nodes. It also decides the service quality and provides reliable and efficient connection for network application nodes. The IP network layer receives the packets sent from the transport layer. Transmission is automatically routed to the address, and packet format can be changed to that of a different protocol, the network traffic condition is monitored, and an overall network topology framework is dynamically built to provide the best route for data transmission. The network access layer receives the packets sent from the IP network layer to divide, combine, and detect the data frame in more detail and correct errors, and ACK is used to determine whether the data is transmitted normally and controls the transfer rate. It is responsible for transmitting the original network data formed between 0 and 1. The method used in this study was TCP/IP. Socket is a port used for network communication in the UNIX system. In general program operation, it can be regarded as an operation for File I/O. Basically, this relies on two key points: read and write. How to read the incoming data and how to write the output. The basic steps before performing these operations are shown in Figure 14.

Experimental results and discussion

Details of the experimental flowchart are shown in Figure 15. An image of the object is acquired by the CCD camera and transferred to the PC via the USB connection. The Hough transformation method is used to analyze and compute the information until the workpiece to be gripped is found as shown in Figure 16. The actual flow is from the original RGB image input (Figure 16(a)) which has been reduced to grayscale (Figure 16(b)) to reduce its computational complexity and the SOBEL level (Figure 16(c)), vertical detection of the image (Figure 16(d)) has been started and integrated (Figure 16(e)), the gradient values have been converted into the Hough space (Figure 16(f)) to determine if it is a circle, and finally, the circle identification (Figure 16(g)) has been completed. The mass point of the object is then found by calculation and the D–H coordinate system transfers this information (by the socket method) to the UR5 robot controller. The robotic arm carries out the necessary movement to grip the workpiece. It automatically moves to the correct position and completes the movement as shown in Figures 17 and 18.

After the workpiece has been gripped, the robot arm receives commands (via Socket) to move to the fixture position where the workpiece is to be placed as shown in Figure 19. The second CCD camera provides
Figure 14. Socket flowchart.

Start

Did not find

The first camera, looking for a specified workpiece through Hough detection

Found it

Return the workpiece coordinate to the robot arm

The robot moves and grips the workpiece

The workpiece is placed in the fixture for processing

Waiting position before the robot arm moves to the clamp

The second camera detects if the workpiece position is accurate

No

Take out after completion

Figure 15. Experiment flowchart.
information for visual identification, and any deviation between the workpiece being held by the arm and the fixture will automatically be corrected. After this, the work continues, and the workpiece is placed smoothly into the fixture. The workpiece is finally picked out of the fixture after machining has been done. The system described here also has a user-friendly man-machine interface for monitoring the operations.

The Hough transformation is used to identify the workpiece to be clamped on the captured image (Figure 17(a)). The robot arm moves to waiting point 1 (Figures 17(b) and 18(a)). After the centroid of the workpiece is found, the coordinate of the workpiece to be selected is sent to the UR5 robot arm using the Socket method, and UR5 robot arm is moved to the first position, point 1. After a wait at point 2 (Figures 17(c) and 18(b)), it finally moves to the workpiece position (Figures 17(d) and 18(c)) to capture the workpiece (Figures 17(e) and 18(d)).

The Hough transformation is useful for recognizing the specific lines of an image by binarization and uses all the pixel points in an image to find the parameters of a straight line. Each point obtains its parameter value by one-to-many projection to the parameter space, and parameter values are generated by all the points to find the most obvious line parameter in the space. The Generalized Hough Transformation does not have much influence on image distortion or pixel loss and is not limited to the recognition of straight lines. In fact, it can be used to recognize any shape. However, it cannot make an efficient classification, and the outcome is not stable enough for computer vision. The parameter space used to search for the target can become very large in a short time, and a lot

![Figure 16. Finding the workpiece using the Hough transformation: (a) original image; (b) binary image from grayscale; (c) horizontal gradient from Sobel; (d) vertical gradient from Sobel; (e) integration of horizontal and vertical Sobel gradients; (f) Hough space; (g) finding the workpiece.](image)
Figure 17. Separate images for gripping: (a) find the workpiece to be gripped; (b) move arm to waiting point 1; (c) move arm to waiting point 2; (d) move arm to workpiece position; (e) grip the workpiece.

Figure 18. Gripping by the robotic arm: (a) move arm to waiting point 1; (b) move arm to waiting point 2; (c) move arm to the workpiece position; (d) grip the workpiece.
of storage space is needed. However, the Hough transformation can solve the above-mentioned problems, and it was chosen for this study.

It can be seen in Table 2 that after improvement, the Hough transformation method can quite accurately locate the mass point of the workpiece to be gripped. The pixel error for the mass center was also reduced by 3.09 pixels. However, the catch rate, when compared to the method before improvement, was reduced by 26.27%. However, the time for image acquisition (before the robotic arm grips the workpiece) is more than adequate for the Hough transformation to determine the mass center of the workpiece to be gripped. Therefore, image recognition, as performed by our improved Hough transformation method, was used in this study.

**Conclusion**

In this study, communication between the “image” and the robotic arm was proposed to reduce preexisting errors in location. Integration of the robotic arm and image identification allowed automatic judgment to be made without the need for an extra manipulator. Correction of the inherent errors and accurate calibration was achieved via feedback vision. This allowed the endpoint pose position of the arm to be fine-tuned, and this significantly improved its accuracy. Measurements were made with the highly stable Hough transformation

![Figure 19. Automatic error calibration using Hough transformation.](image)

| Table 2. Hough transformation method comparison. |
|-----------------------------------------------|
| **Mass center accuracy** | **Image error** | **Image catch rate** |
| Before improvement | 32% | 3.36 pixel | 100% |
| After improvement | 50.7% | 0.27 pixel | 73.73% |
method rather than by the in-depth mechanical learning methods used before. Hough transformation works without the need to build a huge database of training data and saves a lot of time. Results showed that changing the parameters to increase the Hough transformation accuracy not only reduced image pixel error but also increased the precision with which the mass point position could be determined and gave better gripping performance.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study was supported by the Ministry of Science and Technology under Project number: MOST 107–2218-E-167–001.

ORCID iD

Ying-Che Kuo https://orcid.org/0000-0001-9087-5158

References

1. Luo RC and Kuo C-W. Intelligent seven-DoF robot with dynamic obstacle avoidance and 3-D object recognition for industrial cyber–physical systems in manufacturing automation. Proc IEEE 2016; 104: 1102–1113.
2. Guerin KR, Lea C, Paxton C, et al. A framework for end-user instruction of a robot assistant for manufacturing. In: 2015 IEEE international conference on robotics and automation (ICRA), Seattle, WA, 26–30 May 2015, pp.6167–6174. New York: IEEE.
3. Kyrkjebø E, Laastad MJ and Ø S. Feasibility of the UR5 industrial robot for robotic rehabilitation of the upper limbs after stroke. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), Madrid, Spain, 1–5 October 2018, pp.1–6. New York: IEEE.
4. Mathiassen K, Fjellin JE, Glette K, et al. An ultrasound robotic system using the commercial robot UR5. Frontiers in Robotics and AI 2016; 3: 1–16.
5. Atique MMU and Ahad M. Inverse kinematics solution for a 3DOF robotic structure using Denavit-Hartenberg convention. In: 2014 international conference on informatics, electronics & vision (ICIEV), Dhaka, Bangladesh, 23–24 May 2014, pp.1–5. New York: IEEE.
6. Hock O, Drgoňa P and Páskala M. Simulation model of adjustable arm using Denavit-Hartenberg parameters. In: 2014 ELEKTRO, Rajecke Teplice, Slovakia, 19–20 May 2014, pp.176–179. New York: IEEE.
7. Wang H, Qi H, Xu M, et al. Research on the relationship between classic Denavit-Hartenberg and modified Denavit-Hartenberg. In: 2014 seventh international symposium on computational intelligence and design, Hangzhou, China, 13–14 December 2014, pp.26–29. New York: IEEE.
8. Liang B, Cheng Y, Zhu X, et al. Calibration of UR5 manipulator based on kinematic models. In: 2018 chinese control and decision conference (CCDC), Shenyang, China, 9–11 June 2018, pp.3552–3557. New York: IEEE.
9. Kebria PM, Al-Wais S, AbdI H, et al. Kinematic and dynamic modelling of UR5 manipulator. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC), 9–12 October 2016, pp.004229–004234.
10. Liu Z, Chang W, Sheng S, et al. A novel upper limb training system based on UR5 using sEMG and IMU sensors. In: 2016 IEEE international conference on robotics and biomimetics (ROBIO), Qingdao, China, 3–7 December 2016, pp.1069–1074. New York: IEEE.
11. Satzoda RK, Sathyanarayana S, Srikanthan T, et al. Hierarchical additive hough transform for lane detection. IEEE Embedded Syst Lett 2010; 2: 23–26.
12. Shi D, Zheng L and Liu J. Advanced Hough transform using a multilayer fractional Fourier method. IEEE Trans Image Process 2010; 19: 1558–1566.
13. Yokoya N and Iwasaki A. Generalized-Hough-transform object detection using class-specific sparse representation for local-feature detection. In: 2015 IEEE international geoscience and remote sensing symposium (IGARSS), Milan, Italy, 26–31 July 2015, pp.2852–2855. New York: IEEE.
14. Kofman J, Wu XH, Luu TJ, et al. Teleoperation of a robot manipulator using a vision-based human-robot interface. IEEE Trans Indus Electron 2005; 52: 1206–1219.
15. Alambeigi F, Murphy RJ, Basafa E, et al. Control of the coupled motion of a 6 DoF robotic arm and a continuum manipulator for the treatment of pelvis osteolysis. In: 2014 36th annual international conference of the IEEE engineering in medicine and biology society, Chicago, IL, USA, 26–30 August 2014, pp.6521–6525. New York: IEEE.
16. Paxton C, Hundt A, Jonathan F, et al. CoSTAR: instructing collaborative robots with behavior trees and vision. In: 2017 IEEE international conference on robotics and automation (ICRA), Singapore, 29 May–3 June 2017, pp.564–571. New York: IEEE.

17. Jiang QM and Kumar V. The inverse kinematics of cooperative transport with multiple aerial robots. IEEE Trans Robot 2013; 29: 136–145.

18. Shimizu M, Kakuya H, Yoon WK, et al. Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution. IEEE Trans Robot 2008; 24: 1131–1142.

19. Sun J, Cao G, Li W, et al. Analytical inverse kinematic solution using the D-H method for a 6-DOF robot. In: 2017 14th international conference on ubiquitous robots and ambient intelligence (URAI), Jeju, South Korea, 28 June–1 July 2017, pp.714–716. New York: IEEE.

20. Wu L, Crawford R and Roberts J. An analytic approach to converting POE parameters into D–H parameters for Serial-Link Robots. IEEE Robot Autom Lett 2017; 2: 2174–2179.

21. Corke PI. A simple and systematic approach to assigning Denavit-Hartenberg parameters. IEEE Trans Robot 2007; 23: 590–594.

22. Li X, Wang H, Lu X, et al. Neural network method for robot arm of service robot based on D-H model. In: 2017 Chinese automation congress (CAC), Jinan, China, 20–22 October 2017, pp.3273–3277. New York: IEEE.

23. Torres V, Valls J and Canet MJ. Optimised CORDIC-based atan2 computation for FPGA implementations. Electron Lett 2017; 53: 1296–1298.

24. Torres V, Valls J and Lyons R. Fast- and low-complexity atan2(a,b) approximation [tips and tricks]. IEEE Signal Process Mag 2017; 34: 164–169.

25. Lin CJ, Lin CR, Yu SK, et al. Singularity avoidance for a redundant robot using fuzzy motion planning. AMM 2014; 479–480: 729–736.

26. Knierim KL and Sawodny O. Tool-center-point control of the KAI manipulator using constrained QP optimization. Mechatronics 2015; 30: 85–93.

27. Han H and Park J. Robot control near singularity and joint limit using a Continuous Task Transition Algorithm. Int J Adv Robot Syst 2013; 10: 346.

28. Lin CJ, Chen CS, Yu S, et al. A GPU-based evolution algorithm for motion planning of a redundant robot. Int J Adv Robot Syst 2017; 2: 1–14.

29. Donelan PS. Singularity-theoretic methods in robot kinematics. Robotica 2007; 25: 641–659.

30. Lestriandoko NH and Sadikin R. Circle detection based on hough transform and Mexican Hat filter. In: 2016 International conference on computer, control, informatics and its applications (IC3INA), Tangerang, Indonesia, 3–5 October 2016, pp.153–157. New York: IEEE.

31. Lin C-J, Hsu H-H, Cheng C-H, et al. Design of an Image-Servo Mask alignment system using dual CCDs with an XXY stage. Appl Sci 2016; 6(2): 42.

32. Ballard DH. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recogn 1981; 13: 111–122.

33. Seo S and Kim M. Efficient architecture for circle detection using Hough transform. In: 2015 International conference on information and communication technology convergence (ICTC), Jeju, South Korea, 28–30 October 2015, pp.570–572. New York: IEEE.

34. Duda RO and Hart PE. Use of the Hough transformation to detect lines and curves in pictures. Commun ACM 1972; 15: 11–15.

35. Jin S, Kim W and Jeong J. Fine directional de-interlacing algorithm using modified Sobel operation. IEEE Trans Consumer Electron 2008; 54: 857–862.

36. Kim J and Lee S. Extracting major lines by recruiting zero-threshold Canny Edge Links along Sobel highlights. IEEE Signal Process Lett 2015; 22: 1689–1692.

37. Hui-Ping L, Dai-Min C and Miao Y. Communication of multi-robot system on the TCP/IP. In: 2011 International conference on mechatronic science, electric engineering and computer (MEC), Jilin, China, 19–22 August 2011, pp.1432–1435. New York: IEEE.