Hindawi
Journal of Sensors
Volume 2020, Article ID 9270829, 9 pages
https://doi.org/10.1155/2020/9270829

Research Article

The Manipulation of Real-Time Kinect-Based Robotic Arm Using Double-Hand Gestures

Chern-Sheng Lin,[1] **Pei-Chi Chen**,[1] **Yu-Ching Pan**,[1] **Che-Ming Chang**,[2] **and Kuo-Liang Huang**[3]

[1] *Department of Automatic Control Engineering, Feng Chia University, Taichung, Taiwan*
[2] *Ph.D. Program of Electrical and Communications Engineering, Feng Chia University, Taichung, Taiwan*
[3] *Department of Marketing and Supply Chain Management, Overseas Chinese University, Taichung, Taiwan*

Correspondence should be addressed to Chern-Sheng Lin; lincs@fcu.edu.tw

Received 28 April 2019; Revised 15 December 2019; Accepted 2 January 2020; Published 25 January 2020

Academic Editor: Jaime Lloret

Copyright © 2020 Chern-Sheng Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study focused on utilizing the Kinect depth sensor to track double-hand gestures and control a real-time robotic arm. The control system is mainly composed of the microprocessor, a color camera, the depth sensor, and the robotic arm. The Kinect depth sensor was used to take photos of the human body to analyze the skeleton of a human body and obtain the relevant information. Such information was used to identify the gestures of the left hand and the left palm of the user. The gesture of left hand was used as an input command device. The gesture of the right hand was used for imitation movement teaching of robotic arm. From the depth sensor, the real-time images of the human body and the deep information of each joint were collected and converted to the relative positions of the robotic arm. Combining forward kinematics and inverse kinematics and D-H link, the gesture information of the right hand was calculated, which was converted via coordinates into each angle of the motor of the robotic arm. From the color camera, when the left palm was not detected, the user could simply use the right hand to control the action and movement of the real-time robotic arm. When the left palm was detected and 5 fingertips were identified, it meant the start of recording the real-time imitation movement of the robotic arm by the right hand. When 0 fingertip was identified, it meant the stoppage of the above recording. When 2 fingertips were identified, the user could not only control the real-time robotic arm but also repeat the recorded actions.

1. Introduction

The depth sensing techniques have been widely used in the field of virtual reality. A normal measurement system consists of two cameras can obtain the depth values at two different information must be identical because the two cameras look at the same point on the object being measured [1]. The depth sensor can incorporate image processing technology to achieve user and system positioning. The gesture control function and command system enable users to operate the system without wearing complicated and heavy devices [2, 3]. When multiple sensors and cameras were used for positioning, these cameras were set in a sequential group to reduce the complexity of the adjustments and locate the target object at any time [4–6].

The main purpose of this study was to combine a machine vision system, algorithm, and robotic arm to develop an interface system for real-time interactive operation among environment, man, and machine. It also combined image processing technology so as to achieve feasible real-time interactive effects between the system and its user. The machine vision system allows a robotic arm to perceive the external environment. Users can thus employ a robotic arm to perform manual labor in a harsh environment.

First of all, through the Kinect sensor, the depth, width, and color of the user in the 3D space were captured. OpenCV and Visual C++ were adopted to establish the human skeleton data and analyze the deep information on the user's joints. The information on the joints of the right hand was converted into the relative coordinate system of the robotic arm. In combination with inverse kinematics and D-H link, the angle of rotation required for each joint of the robotic arm was calculated, which was transferred to the Arduino controller for the real-time operation of the robotic arm by the user's right hand. In terms of the recognition of the user's left-hand gestures, the Kinect sensor was used to track the human skeleton so as to determine the positions of the hand. According to the location of the hand, the adjacent depth was set, and the area of the hand was captured. Through OpenCV, the outline of the hand was identified, followed by the convex and concave points of the hand. Such convex and concave points were then analyzed to determine the left-hand gestures.

Recently, there are many related researches on object recognition and Kinect sensing devices of different visual systems. For examples, FEMD (Finger-Earth Mover's Distance) is an algorithm specially designed for the shape of the hand. This method mainly removes the palm part and processes the end of the finger. At this time, Kinect's sensor can get better hand shape [7]. Shum et al. focused on image reconstruction when the human body is obscured. They used Kinect to recognize human posture. Based on the reliability assessment, it performed an action database query [8]. Chang et al. performed their experiments on three public Kinect-based entertainment data sets related to fitness and gaming [9]. Lloret et al. present a wireless sensor network where each sensor node takes images from the field and internally uses image processing techniques to detect any unusual status in the leaves [10]. A sequential flow of moving object detection can track and identify the behavior completes the processing framework of video surveillance [11].

This paper used the Kinect sensor to extract the gestures and depth information of the current user. Through image processing and tracking algorithm, the coordinate positions of the hands and joints of the current user were calculated. Through gestures with image separation and template matching [12, 13], the left palm was identified. The gesture of left hand was used as an input command device. The gesture of the right hand was used for imitation movement teaching of robotic arm. When the left palm was detected and 5 was identified, it meant the start of recording the real-time control of the robotic arm by the right hand. When 0 was identified, it meant the stoppage of the above recording. The right palm could record the real-time control of the robotic arm by the right hand. The user could not only control the real-time robotic arm but also repeat the recorded actions. The gestures of the right hand were then tracked (Figure 1). Through inverse kinematics, the coordinate positions of the joints of the right hand were estimated [14]. Such positions were converted to the coordinate system relative to the positions of the robotic arm. Inverse kinematics was used to estimate the angle of rotation of each joint of the robotic arm required for the motor. The Arduino controller was then used to control the motor on each joint so as to achieve real-time manipulation of the robotic arm by the user.

2. The Multiple Sensing Process

This paper used the Kinect sensor to capture the coordinate position of each joint of the user; the paper converted the coordinate system of the user into that of the AL5D robotic arm through a proportional conversion relationship. The control system is mainly composed of the microprocessor, a color camera, the depth sensor, and the robotic arm.

From the depth sensor, the real-time images of the human body and the deep information of each joint were collected and converted to the relative positions of the robotic arm. For example, in consideration of the original point of the robotic arm and the coordinate axis directions, the distance between the original point of the arm and the shoulder was $d_1 = 7\,\text{cm}$. The distance between the shoulder and the elbow was $a_2 = 14.605\,\text{cm}$, that between the elbow and the wrist was $a_3 = 18.7325\,\text{cm}$, and that between the wrist and the end of the palms was $a_4 = 10.0076\,\text{cm}$. As the coordinates of the shoulder of AL5D robotic arm were not to be changed, the coordinates x, y, and z of the shoulder were set as 0, 0, and 7, respectively. The coordinates of the elbow and arms would change along with the different gestures of the user (Figure 2).

Kinect was used to measure the coordinates of the shoulder $(x_1{}', y_1{}', z_1{}')$ and arms $(x_2{}', y_2{}', z_2{}')$. The length of the link between the shoulder and elbow was calculated with the following formula:

$$D_1 = \sqrt{\left(x_2' - x_1'\right)^2 + \left(y_2' - y_1'\right)^2 + \left(z_2' - z_1'\right)^2}. \tag{1}$$

The length of the link between the shoulder and elbow was 7 cm. Hence, the proportional relationship a_2/D_1 between the lengths of the two links was as follows:

$$\frac{a_2}{D_1} = \frac{7}{\sqrt{\left(x_2' - x_1'\right)^2 + \left(y_2' - y_1'\right)^2 + \left(z_2' - z_1'\right)^2}}. \tag{2}$$

In order to calculate the coordinate changes from the shoulder to elbow of the AL5D robotic arm, this paper sets the coordinate changes of the shoulder and elbow as $(x_2{}' - x_1{}', y_2{}' - y_1{}', z_2{}' - z_1{}')$, which were then multiplied with the proportional relationship between the shoulder and elbow of the AL5D robotic arm. The result was the coordinate changes from the shoulder to elbow of AL5D robotic arm:

$$\frac{a_2}{D_1} \bullet \left(x_2' - x_1', y_2' - y_1', z_2' - z_1'\right). \tag{3}$$

The coordinates of the elbow of the AL5D robotic arm were set to (0, 0, 7), while those of the elbow were (x_2, y_2, z_2). The coordinates of the shoulder of the users plus the coordinates changes from the shoulder to elbow of AL5D robotic arm resulted in the coordinates of the elbow of AL5D robotic arm (x_2, y_2, z_2), as follows:

$$(x_2, y_2, z_2) = (0, 0, 7) + \frac{a_2}{D_1} \bullet \left(x_2' - x_1', y_2' - y_1', z_2' - z_1'\right). \tag{4}$$

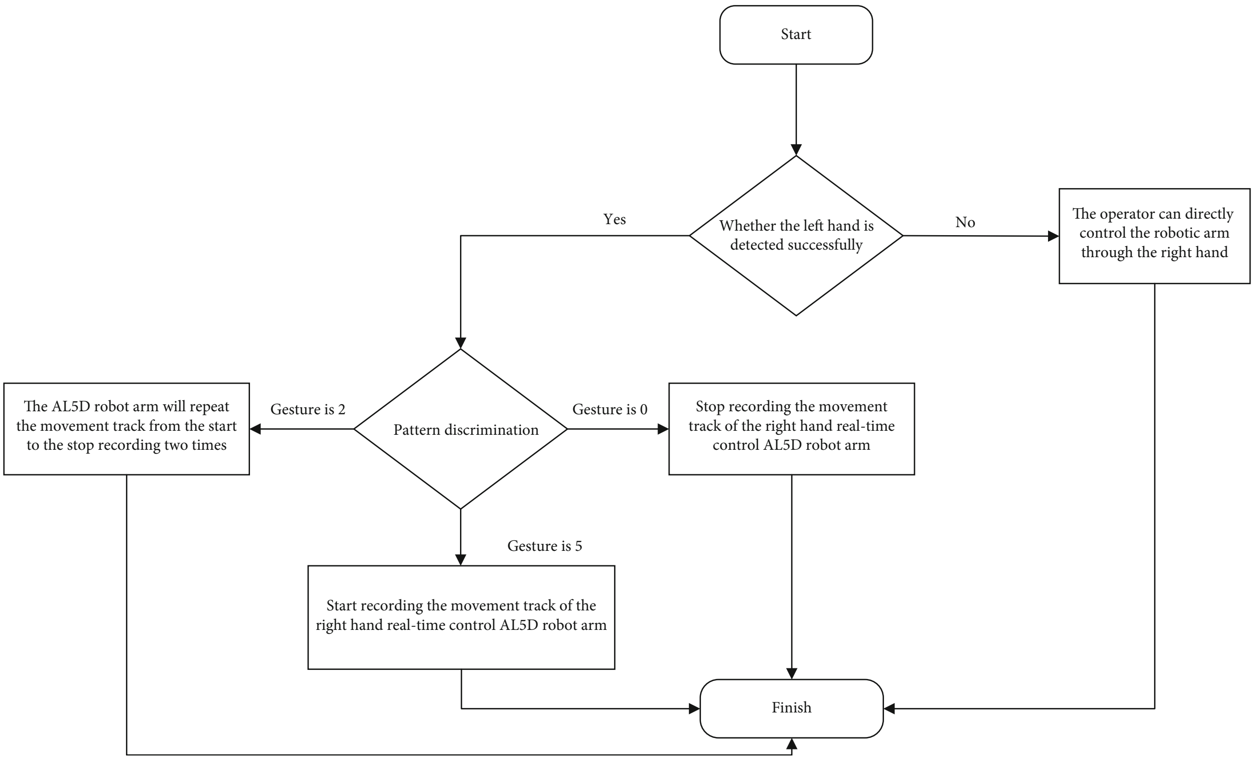

Start
Whether the left hand is detected successfully
Yes
No
The operator can directly control the robotic arm through the right hand
Pattern discrimination
Gesture is 2
Gesture is 0
Gesture is 5
The AL5D robot arm will repeat the movement track from the start to the stop recording two times
Stop recording the movement track of the right hand real-time control AL5D robot arm
Start recording the movement track of the right hand real-time control AL5D robot arm
Finish

Figure 1: The control process and the main structure of this study.

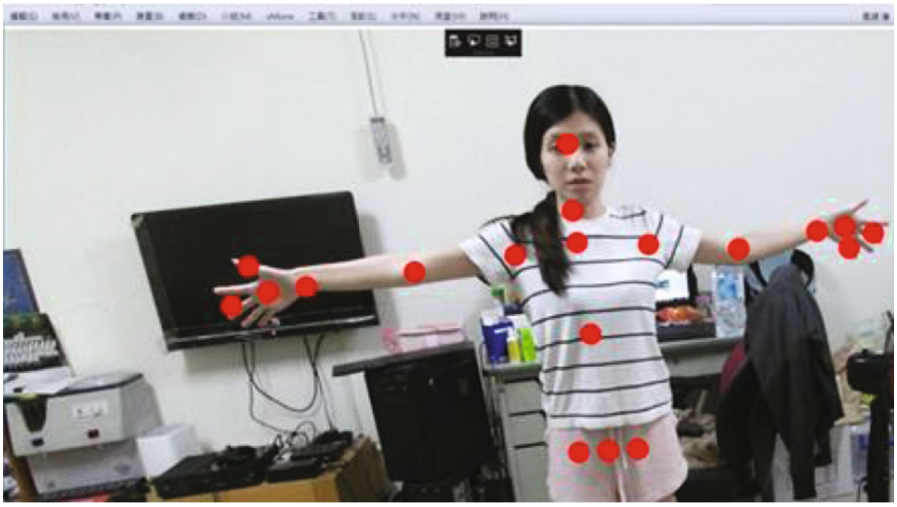


Figure 2: The coordinates of the joints of the upper body detected by the Kinect sensor.

After the coordinates (x_2, y_2, z_2) of the elbow of AL5D robotic arm were calculated, for the purpose of the inverse kinematics calculation later, the coordinates of the wrist should be obtained. Thus, the elbow coordinates (x_3, y_3, z_3) of the AL5D robotic arm should be calculated.

Set the wrist coordinates of the user as (x_2', y_2', z_2') and the elbow coordinates as (x_3', y_3', z_3'). The length of the link between the elbow and wrist was calculated as follows:

$$D_2 = \sqrt{\left(x_3' - x_2'\right)^2 + \left(y_3' - y_2'\right)^2 + \left(z_3' - z_2'\right)^2}. \qquad (5)$$

The length of the link between the shoulder and elbow of the robotic arm was 18.7325 cm. Hence, the proportional relationship a_3/D_2 between the lengths of the two links was as follows:

$$\frac{a_3}{D_2} = \frac{18.7325}{\sqrt{\left(x_3' - x_2'\right)^2 + \left(y_3' - y_2'\right)^2 + \left(z_3' - z_2'\right)^2}}. \qquad (6)$$

In order to calculate the coordinate changes from the elbow to wrist of the AL5D robotic arm, this paper sets the coordinate changes of the elbow and wrist of the user as $(\mathrm{x}_3{}' - \mathrm{x}_2{}', \mathrm{y}_3{}' - \mathrm{y}_2{}', \mathrm{z}_3{}' - \mathrm{z}_2{}')$, which were then multiplied with the proportional relationship between the elbow and wrist of the AL5D robotic arm. The result was the coordinate changes from the elbow to wrist of the AL5D robotic arm:

$$\frac{a_3}{D_2} \bullet \left(x_3' - x_2', y_3' - y_2', z_3' - z_2'\right). \tag{7}$$

The coordinates of the elbow of the AL5D robotic arm were set to (x_2, y_2, z_2), while those of the wrist were (x_3, y_3, z_3). The coordinates of the elbow of the users (x_2, y_2, z_2) plus the coordinates changes from the elbow to wrist of the AL5D robotic arm resulted in the coordinates of the wrist of AL5D robotic arm (x_3, y_3, z_3), as follows:

$$(x_3, y_3, z_3) = (x_2, y_2, z_2) + \frac{a_3}{D_2} \bullet \left(x_3' - x_2', y_3' - y_2', z_3' - z_2'\right). \tag{8}$$

After the coordinates (x_3, y_3, z_3) of the elbow of the AL5D robotic arm were calculated, for the purpose of the inverse kinematics calculation later, the coordinates of the wrist should be obtained. Thus, the coordinates (x_4, y_4, z_4) of the end of the palm of the AL5D robotic arm should be calculated.

Set the wrist coordinates of the user as $(x_3{}', y_3{}', z_3{}')$ and the coordinates of the end of palm as $(x_4{}', y_4{}', z_4{}')$. The length D3 of the link between the wrist and end of palm was calculated as follows:

$$D_3 = \sqrt{\left(x_4' - x_3'\right)^2 + \left(y_4' - y_3'\right)^2 + \left(z_4' - z_3'\right)^2}. \tag{9}$$

The length of the link between the wrist and the end of the palm of the AL5D robotic arm was 10.0076 cm. Hence, the proportional relationship a_4/D_3 between the lengths of the two links was as follows:

$$\frac{a_4}{D_3} = \frac{10.0076}{\sqrt{\left(x_4' - x_3'\right) + \left(y_4' - y_3'\right) + \left(z_4' - z_3'\right)}}. \tag{10}$$

In order to calculate the coordinate changes from the wrist to the end of the palm of the AL5D robotic arm, this paper sets the coordinate changes of the wrist and the end of the palm of the user as $(x_4{}' - x_3{}', y_4{}' - y_3{}', z_4{}' - z_3{}')$, which were then multiplied with the proportional relationship between the wrist and the end of the palm of the AL5D robotic arm. The result was the coordinate changes from the wrist to the end of the palm of the AL5D robotic arm:

$$\frac{a_4}{D_3} \bullet \left(x_4' - x_3', y_4' - y_3', z_4' - z_3'\right). \tag{11}$$

The coordinates of the wrist of the AL5D robotic arm were set to (x_3, y_3, z_3), while those of the end of the palm were (x_4, y_4, z_4). The coordinates of the wrist of the users (x_3, y_3, z_3) plus the coordinates changes from the wrist to the end of the palm of the AL5D robotic arm resulted in the coordinates of the end of the palm of the AL5D robotic arm (x_4, y_4, z_4), as follows:

$$(x_4, y_4, z_4) = (x_3, y_3, z_3) + \frac{a_4}{D_3} \bullet \left(x_4' - x_3', y_4' - y_3', z_4' - z_3'\right). \tag{12}$$

In terms of direct kinematics, the Denavit-Hartenberg (D-H) rule is widely used to define a robotic arm (Figure 3). It originally took six parameters (horizontal movement along and the rotation around the three axes) to describe the motion or location relationship of a robotic arm. With the D-H coordinate system, only four are required, which simplifies mathematical operations and makes the D-H rule a standard to describe the kinematics of a robot.

This paper used the Denavit-Hartenberg representation proposed in 1955 to set up the coordinate system. The coordinate system of each axis joint was defined according to the following rules. The parameters of the four links [7, 8] and the definitions and diagrams of each link were also obtained.

(i) Link length a_i: calculate the distance from Z_{i-1} to Z_i along X_i

(ii) Link twist α_i: calculate the distance from Z_{i-1} to Z_i along X_i

(iii) Link offset d_i: calculate the distance from X_{i-1} to X_i along Z_{i-1}

(iv) Joint angle θ_i: calculate the distance from X_{i-1} to X_i along Z_{i-1}

The final result of the transformation matrix $T^i(i-1)$ can be obtained from the transformation matrix:

$$\begin{gathered} T_i^{i-1} = \operatorname{Trans}(z, d_i)\operatorname{Rot}(z, \theta_i)\operatorname{Trans}(x, a)\operatorname{Rot}(x, \alpha), \\ T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & \alpha_i \cos\theta_i \\ \sin\theta & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & \alpha_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{gathered} \tag{13}$$

Denavit-Hartenberg (D-H) conversion matrix [9].

From the color camera, the fingertips of the image of two palms can be detected with convex set method. In a real vector space V, for the given set X, all the intersection S containing the convex set X were named

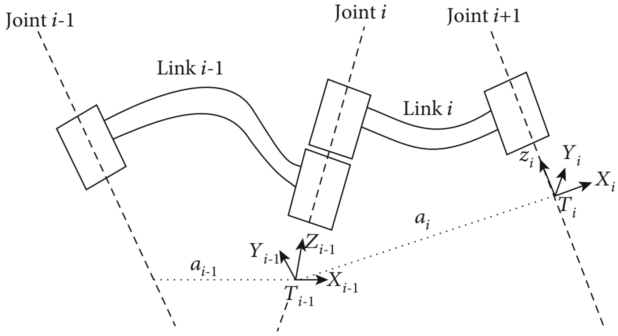


Figure 3: Denavit-Hartenberg representation.

the convex hull of X. The set mathematical notation is Equation (14) [15–17].

$$S \coloneqq \bigcap_{\substack{X\subseteq K\subseteq V \\ K \text{ is convex}}} K. \tag{14}$$

The definition of "convex" is the link between any two points in an image does not pass through the external part of the image. "Convex" does not refer to the bulged surface. In fact, convex hull is made up of many flat surfaces [18, 19]. The convex hull algorithm was used for fingertip detection. The concept of the algorithm lies in assuming a point was set in a 2D surface; all the peripheral points were connected to one point to cover the point set. If there were overlapping of several points in the convex hull, in the context of multipoint and commonline, then generally the least points would be used to judge the result, i.e., the minimum area [18, 20].

The convex hull of X could be constructed by the linear combination of all the points $(X_1, \cdots, X_n)$ in X [21, 22].

$$S \coloneqq \left\{ \sum_{j=1}^{n} t_j x_j \mid x_j \in X,\ \sum_{j=1}^{n} t_j = 1, \in [0,1] \right\}. \tag{15}$$

This study used the convex hull algorithm to identify the left-hand gestures so that Kinect could determine the left-hand gesture mode. First, Kinect was used to track the human skeleton and the left-hand joints and search between the two joints of HandLeft and HandTipLeft. The depth outside the scope was excluded so as to find the contour of the hand. As the convex hull algorithm could effectively find the five points corresponding to fingertips, the fingertips were the edges of all the contour polyline, and the edges were the fingers. The convex hull algorithm could be employed to analyze these points and then determine the left-hand gestures.

First up was the identification of the left-hand gestures. The Kinect sensor was adopted to track the human skeleton [23–25]. Second, the left-hand joints were detected. Third, the search area was specified. When the left-hand joints were detected, search was made within a reasonable scope of the HandLeft joints. The reasonable search scope was restricted between the joints of HandLeft and HandTipLeft.

Fourth, find the contour of the hands. As Step 3 strictly defined the search area in the 3D space, any depth values outside the scope could be excluded. Thus, all the depth values outside the hands would be rejected. Thus, the shape of a hand could be obtained, and the contour of the shape was that of the hand.

Lastly, find the convex hull. The contour of the hand was a big set. However, only five points (or less) corresponded to the tip. The tip was the edge including all the contour polyline. With the above steps, the left hand could be tracked, as shown in Figure 4. Next, these points could be analyzed to determine hand gestures.

The flowchart to track the right-hand gestures was next. First, the Kinect sensor was adopted to obtain color and deep image data. Next, the location of the human body was detected, and the skeleton of the right hand was detected and tracked so as to find the joints of the right hand. Finally, it was the conversion of coordinates. The coordinates of the right hand of the user were converted into those of the robotic arm. The D-H rule was used to deduce direct kinematics and obtain the coordinates of the end effector (p_x, p_y, p_z), which could be used to deduce inverse kinematics. With the known absolute coordinates of each link, the $\theta_1, \theta_2, \theta_3$, and θ_4 positions of each joint were obtained, which were transmitted to the Arduino controller to drive the robotic arm.

3. Experimental Results

This paper identified five gestures of the left hand. The gesture identification included 5, 0, 2, gestures not detected, and unknown gestures, as shown in Figure 5.

Because of a mirror reflection relation, the following figure shows the right hand. The main requirement here is to take the position of the robot arm end point in the Cartesian coordinate system (x, y, z) and the angle $(\theta_1, \theta_2, \theta_3)$ of the joint coordinate system of the robot arm Conversion relationship between. Calculate the position of the end point of the arm in the Cardison coordinate system by using the known movement angles $(\theta_1, \theta_2, \theta_3)$ of each axis joint, and then infer each position from the known position of the end point of the arm in the Cardison coordinate system. It can be calculated how many degrees does the shaft joint need to move to reach this position. Mapping from the end point of the robot arm to the movement angle of each axis is inverse kinematics, while mapping from the movement angle of each axis to the end point of the robot arm is forward kinematics. When the left gesture was detected and 5 was identified, it meant the start of the recording the real-time control of the robotic arm by the right hand. When 0 was identified, it meant the stoppage of the above recording. When 2 was identified, the robotic arm repeated the motion twice from the start to the end of the record. When the left-hand gesture was not detected, the right hand could manipulate the real-time robotic arm (Figure 6). When unknown gestures were detected, the robotic arm had no action.

The tracking of the right-hand gestures was for the real-time manipulation of the robotic arm. Due to the mirror, the following figure shows the left hand. The Kinect sensor

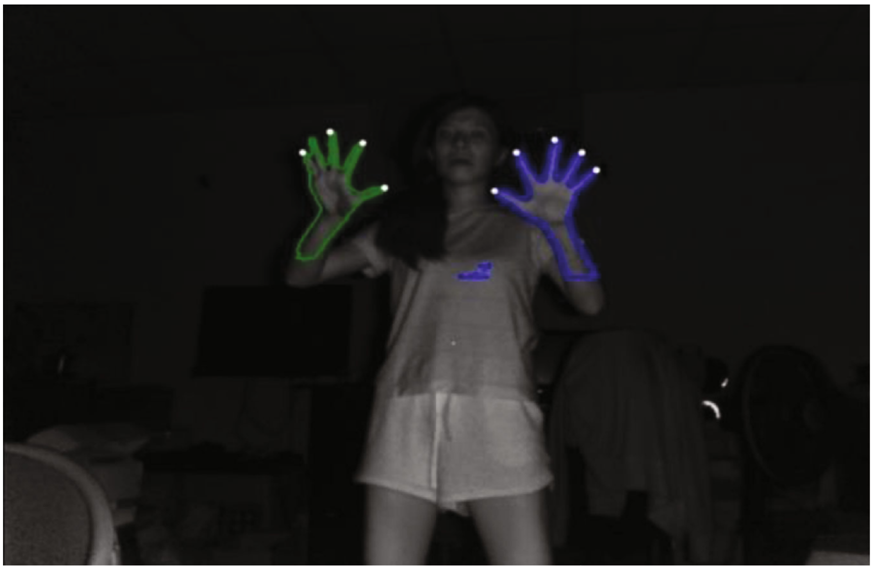
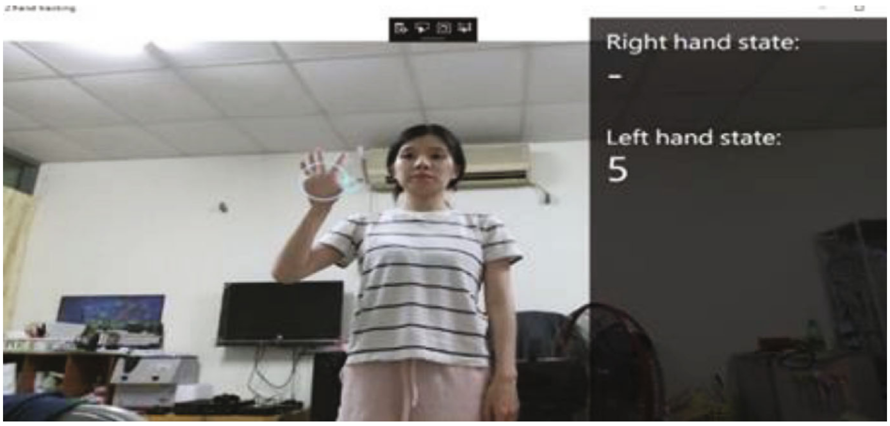


Figure 4: Tracking both hands.



Right hand state:
-

Left hand state:
5

Figure 5: Left hand was identified as 5.

was used to obtain the information on the skeleton of the right hand. Through the conversion of coordinates and inverse kinematic operation, the angle of rotation required for each joint of the robotic arm was obtained (Figure 7). The angles were then transmitted to Arduino, which drove the robotic arm.

Table 1 shows the experimental data on the identification of the left-hand gestures from different distances (100 cm, 150 cm, 200 cm) and different locations and directions so as to reach accuracy.

Each joint of the user's right hand stretched to 90 degrees. Each motor measured the actual movement angle of the robotic arm for three times. Here, the forward and reverse kinematics are used to calculate and verify the angle of the joint motor of each position of the robot arm. Let the user's right hand move 90 degrees in each joint and measure the actual movement angle of the robot arm three times in each motor (Table 2).

It can be known from Table 2 that although the actual movement angles have mutual errors, they can basically operate according to the original instructions.

4. Discussion

There were errors in the actual moving angles; basically, the robotic arm could follow the instructions to conduct direct and inverse kinematic calculation. Moreover, the angle of joint motor of each gesture was verified. The gesture information of the right hand was calculated, which was converted via coordinates into each angle of the motor of the robotic arm. Lastly, the program transmitted the information to microprocessor, which further transmitted it to the corresponding motor of robotic arm, so that the user could control the real-time robotic arm.

The Kinect sensor was employed to identify and track the gestures of both hands. Five gestures of the left hand were identified, including 5, 0, 2, gestures undetected, and unknown gestures. When the left-hand gesture was not detected, the right hand could manipulate the real-time robotic arm. When the left gesture was detected and 5 was identified, it meant the start of the recording the real-time control of the robotic arm by the right hand. When 0 was

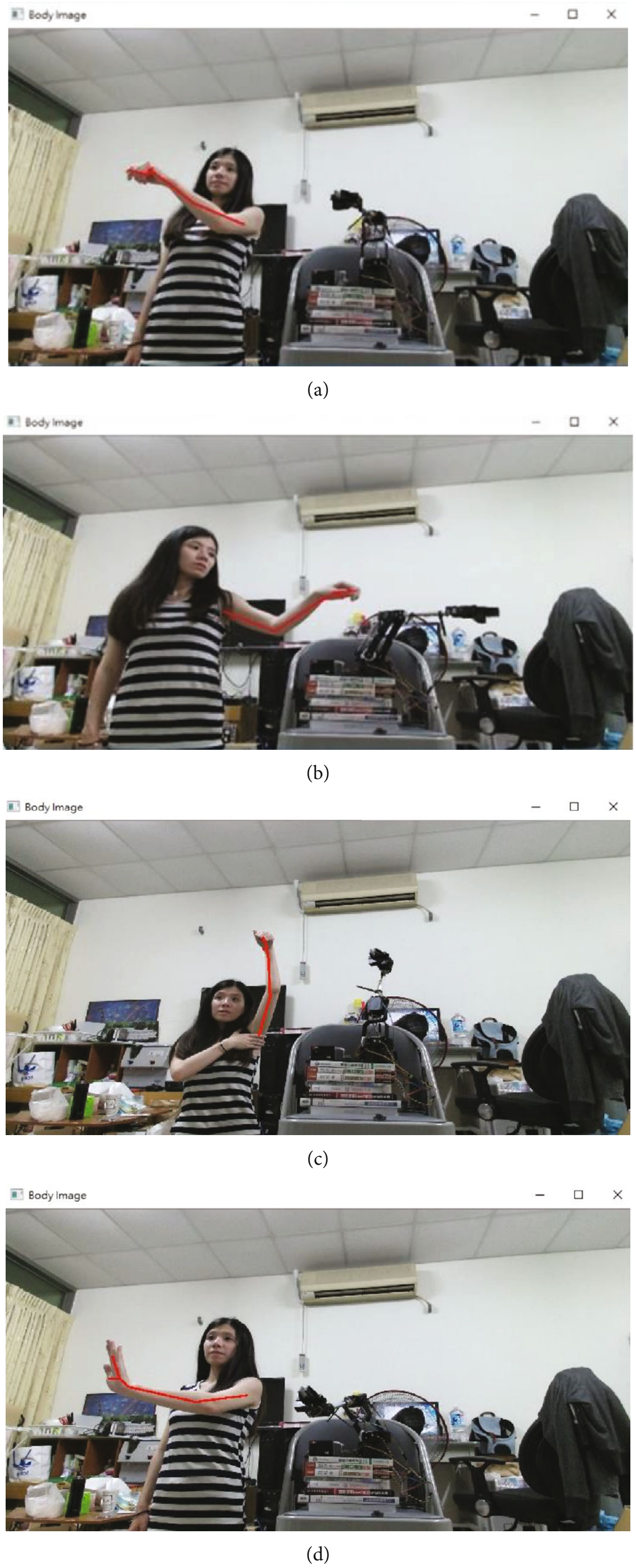
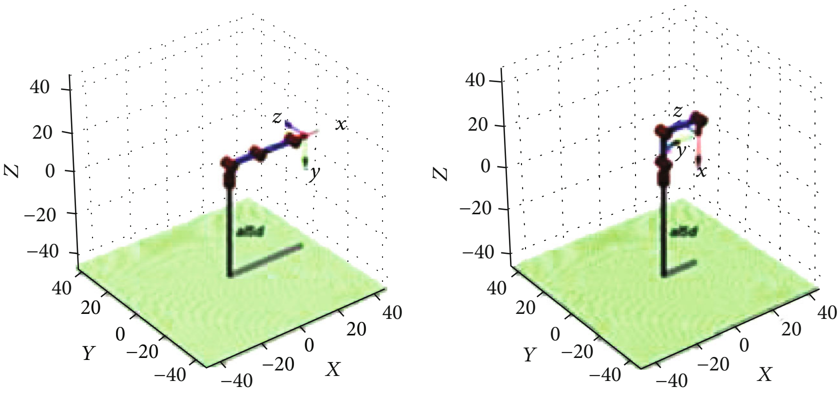


Figure 6: (a) The user used her right hand to manipulate the robotic arm. (b) The user pushed her hand right and forward. (c) The user pushed her right hand upward and bent her wrist downward. (d) The user bent her right hand left slightly and bent her wrist upward.



(a)

(b)

Figure 7: The relationship of coordinates and inverse kinematic operation $[\theta_1\ \theta_2\ \theta_3\ \theta_4]$ = (a) [0 0 0 0] and (b) $[0 - (\pi/2)\ (\pi/2)\ (\pi/2)]$.

Table 1: Experimental data on the identification of the left-hand gestures.

	Quantity of samples	Undetected	Unknown	Count of failures	Accuracy
Gesture was 5	150	0	4	4	97.33%
Gesture was 0	150	0	0	0	100%
Gesture was 2	150	0	3	3	98%

Table 2: The tracking accuracy of the robot.

Joint	Actual angle of robot arm movement (degrees)		
1	104	102	99
2	99	103	102
3	98	98	102
4	92	96	91

identified, it meant the stoppage of the above recording. When 2 was identified, the robotic arm repeated the recording twice.

5. Conclusions

The main purpose of this study was to combine a machine vision system, algorithm, and robotic arm to develop an interface system for real-time interactive operation among environment, man, and machine. It also combined image processing technology so as to achieve feasible real-time interactive effects between the system and its user. The machine vision system allows a robotic arm to perceive the external environment. Users can use a robotic arm to perform manual labor in a harsh environment.

This study tracked the gestures of both hands to realize real-time manipulation of a robotic arm. The Kinect sensor was used to detect the gesture and keep information of the current user. The user could control the actions of the robotic arm though the current images. When the user moved, the D-H rule and the direct and inverse kinematics allowed the robotic arm to perform the same action as that of the user, resulting in the effect of interaction.

In this paper, gesture recognition and tracking of both hands are performed through a Kinect sensor. Left-hand gesture recognition can recognize five gestures, which are 5, 0, 2, no gesture detected, and unknown gesture. Through the recognition of left-hand gestures, when no gesture is detected, the right hand can control the AL5D robotic arm in real time; when a gesture is detected as 5, it starts to record the trajectory of the right-handed real-time control robotic arm; then stop recording; when the gesture detection is 2, the robot will repeat the recorded track twice.

The color and depth data of the user are mainly obtained through the Kinect sensor to analyze the posture of the left and right hands. The left hand is mainly used to determine the mode that the robot arm must perform. By operating in different modes, the robot arm functions more effectively. It is versatile. The right hand is mainly used to operate the robotic arm. Kinect obtains the user's right hand motion trajectory and substitutes inverse kinematics to obtain the angle of rotation of each axis motor, which has reached the task of driving the robotic arm.

In terms of future development, more gesture recognition can be performed for the left hand, so that the left hand can use more modes to control the robotic arm. At the same time, a camera can be installed on the robotic arm. With the assistance of images, the fuselage can obtain environmental information, analyze its environment, and make future operations more humane.

Data Availability

The theoretical data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflict of interest regarding the publication of this paper.

Acknowledgments

The manuscript is based on the thesis titled "The Manipulation of Real-Time Kinect-Based Robotic Arm Using Double-Hand Gestures" supported by the projects Grant No. MOST 106-2221-E-035-053-MY2 and MOST 107-2321-B-035-001 funded by the Ministry of Science and Technology (MOST), Taiwan.

References

[1] H. Lee and H. Cho, "Stereo Moiré technique, a novel 3-D measurement method using a stereo camera and a digital pattern projector," *International Journal of Optomechatronics*, vol. 1, no. 2, pp. 209–230, 2007.

[2] J. Sun, Z. Liu, Y. Zhao, Q. Liu, and G. Zhang, "Motion deviation rectifying method of dynamically measuring rail wear based on multi-line structured-light vision," *Optics & Laser Technology*, vol. 50, no. 3, pp. 25–32, 2013.

[3] C. S. Lin, C. T. Chen, T. C. Wei, W. L. Chen, and C. C. Chang, "A positioning model of a two CCD camera coordinate system with an alternate- four-matrix look-up table algorithm," *Optics and Lasers in Engineering*, vol. 48, no. 12, pp. 1193–1199, 2010.

[4] C. S. Lin, T. C. Wei, A. T. Lu, S. S. Hung, W. L. Chen, and C. C. Chang, "A rehabilitation training system with double-CCD camera and automatic spatial positioning technique," *Optics and Lasers in Engineering*, vol. 49, no. 3, pp. 457–464, 2011.

[5] C. S. Lin, A. T. Lu, Y. C. Hsu, C. L. Tien, D. C. Chen, and N. C. Chang, "A large-panel two-CCD camera coordinate system with an alternate-eight-matrix look-up-table method," *Optics & Laser Technology*, vol. 44, no. 2, pp. 471–475, 2012.

[6] C. S. Lin, K. S. Chen, Y. C. Huang, and Y. C. Lin, "Multicamera interface with novel copositioning," *Instrumentation Science & Technology*, vol. 44, no. 1, pp. 35–45, 2016.

[7] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[8] H. P. H. Shum, E. S. L. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for Microsoft Kinect," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1357–1369, 2013.

[9] X. Chang, Z. Ma, M. Lin, Y. Yang, and A. G. Hauptmann, "Feature interaction augmented sparse learning for fast Kinect motion detection," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3911–3920, 2017.

[10] J. Lloret, I. Bosch, S. Sendra, and A. Serrano, "A wireless sensor network for vineyard monitoring that uses image processing," *Sensors*, vol. 11, no. 6, pp. 6165–6196, 2011.

[11] S. Ojha and S. Sakhare, "Image processing techniques for object tracking in video surveillance- a survey," in *2015 International Conference on Pervasive Computing (ICPC)*, pp. 1–6, Pune, India, January 2015.

[12] S. W. Yang, C. S. Lin, S. K. Lin, and Y. C. Tseng, "Automatic inspection system for defects of printed art tile based on texture feature analysis," *Instrumentation Science & Technology*, vol. 42, no. 1, pp. 59–71, 2014.

[13] O. K. Kwon, "Accurate slide positioning using fast adaptive-template matching in an automated microscope," *Instrumentation Science & Technology*, vol. 42, no. 4, pp. 385–401, 2014.

[14] S. Amatya and S. Petchartee, "Real time Kinect based robotic arm manipulation with five degree of freedom," in *2015 Asian Conference on Defence Technology*, vol. 45, pp. 742–753, Hua Hin, Thailand, April 2015.

[15] A. A. Mohammed and M. Sunar, "Kinematics modeling of a 4-DOF robotic arm," in *IEEE International Conference on Control, Automation and Robotics*, Singapore, Singapore, May 2015.

[16] Y. K. Fang, J. Cheng, K. Q. Wang, and H. Q. Lu, "Hand gesture recognition using fast multi-scale analysis," in *Fourth International Conference on Image and Graphics(ICIG 2007)*, pp. 50–67, Sichuan, China, August 2007.

[17] W. Du and H. Li, "Vision based gesture recognition system with single camera," in *WCC 2000 - ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000*, pp. 351–1357, Beijing, China, August 2000.

[18] S. M. Nadgeri, S. D. Sawarkar, and A. D. Gawande, "Hand gesture recognition using Camshift algorithm," in *2010 3rd International Conference on Emerging Trends in Engineering and Technology*, pp. 37–41, Goa, India, November 2010.

[19] B. Zheng, L. Zhao, and Y. Wang, "Fingertip detection and gesture recognition based on Kinect depth data," *Transactions on Computer Science and Technology*, vol. 3, pp. 9–14, 2014.

[20] J. M. Palacious, C. Sagiies, E. Montijano, and S. Liorente, "Human-computer interaction based on hand gestures using RGB-D sensors," *Sensors*, vol. 13, no. 9, pp. 11842–11860, 2013.

[21] S. S. Rautaray and A. Agrawal, "Real time hand gesture recognition system for dynamic applications," *International Journal of UbiComp*, vol. 3, no. 1, pp. 21–31, 2012.

[22] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, pp. 511–518, Kauai, HI, USA, December 2001.

[23] M. Ju, Y. Choi, J. Seo et al., "A Kinect-based segmentation of touching-pigs for real-time monitoring," *Sensors*, vol. 18, no. 6, article 1746, 2018.

[24] S. Yun, J. Choi, and C. S. Won, "Omnidirectional 3D point clouds using dual Kinect sensors," *Journal of Sensors*, vol. 2019, Article ID 6295956, 17 pages, 2019.

[25] J. Lee, D.-W. Kim, C. Won, and S.-W. Jung, "Graph cut-based human body segmentation in color images using skeleton information from the depth sensor," *Sensors*, vol. 19, no. 2, p. 393, 2019.