

Development of Multi-Robotic Arm System for Sorting System Using Computer Vision

Vo Duy Cong^{1,4,*}, Dang Anh Duy^{2,4}, Le Hoai Phuong^{3,4}

^{1,2,3} Industrial Maintenance Training Center, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

⁴ Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam

Email: ¹ congvd@hcmut.edu.vn, ² aduy@hcmut.edu.vn, ³ lhphuong@hcmut.edu.vn

*Corresponding Author

Abstract— This paper develops a multi-robotic arm system and a stereo vision system to sort objects in the right position according to size and shape attributes. The robotic arm system consists of one master and three slave robots associated with three conveyor belts. Each robotic arm is controlled by a robot controller based on a microcontroller. A master controller is used for the vision system and communicating with slave robotic arms using the Modbus RTU protocol through an RS485 serial interface. The stereo vision system is built to determine the 3D coordinates of the object. Instead of rebuilding the entire disparity map, which is computationally expensive, the centroids of the objects in the two images are calculated to determine the depth value. After that, we can calculate the 3D coordinates of the object by using the formula of the pinhole camera model. Objects are picked up and placed on a conveyor branch according to their shape. The conveyor transports the object to the location of the slave robot. Based on the size attribute that the slave robot receives from the master, the object is picked and placed in the right position. Experiment results reveal the effectiveness of the system. The system can be used in industrial processes to reduce the required time and improve the performance of the production line.

Keywords— Robot arm; Computer vision; Stereo vision; Modbus RTU; RS485.

I. INTRODUCTION

Nowadays, robotic arms are widely used industrial applications. It can replace human labor in most jobs that can be characterized as repetitive or require a lot of lifting force. The robot can work continuously without fatigue, having high repeatability and reliability and creating high-precision and high-quality products [1-6]. Therefore, the requirements for flexibility, accuracy and robustness of robots are increasing to be able to operate in many different environments. To meet such requirements, robots are often integrated with sensors to measure and sense the surrounding environment (especially when the robot operates in an unstructured environment). The sensors that are commonly used are force sensors, ultrasonic sensors, lasers, vision sensors, etc. Among them, visual sensors (e.g., cameras) are the most common because the way vision sensors perceive the environment is close to that of humans and allows for non-contact measurements [7-11].

The robotic system integrated with a vision system has enabled robots to accomplish many applications as assembly and disassembly, vision guide robot, mapping, navigation,

tracking, path planning, robot localization, exploration, surveillance, search, recognition, inspection [12-20].

Robots can work independently or can be combined in production lines to perform complex tasks involving many stages [21-24]. In a robotic system, robots can be pre-programmed to perform a repetitive action without interacting with other robots or the rest of the system. However, to be able to perform complex tasks and replace human labour, robots must interact with each other. With the continuous evolvement of digital control, new advances in robotics as well as networked strategies being introduced in the industrial scene, robots can be connected and communicate with each other leading to the introduction of collaborative environments and Human-Robot Collaboration (HRC) [13].

In this paper, a multi-robot system is developed that uses a stereo vision system for sorting products in a production line. The robotic arm system consists of one master and three slave robots associated with three conveyor belts. Data is transmitted from the master robot to the slave robots based on the Modbus RTU protocol through an RS485 serial interface. The stereo vision system is built to determine the 3D coordinates of the object. Instead of rebuilding the entire disparity map, which is computationally expensive, the centroids of the objects in the two images are calculated to determine the depth value. After that, we can calculate the 3D coordinates of the object by using the formula of the pinhole camera model. The contribution of this paper is: (1) propose a simple stereo vision system to calculate the 3D position of objects with a low computational cost that can be implemented on a low-cost embedded computer, (2) present the way to communicate between robots through an RS485 network built on the embedded system.

II. RELATED WORKS

A computer vision system can make a robot move more flexibly and faster. They can improve production efficiency and ensure product accuracy [25-29]. For example, industrial robots have been commonly used in industrial production for sorting [30-34]. Wu et al. [31] use a deep convolutional neural network to locate and recognize complex workpieces for the vision-based sorting robot in the industrial production process. The pixel projection algorithm (PPA) is presented to eliminate uneven illumination, located and segment workpieces in the image. DCNN is applied to recognize



rational degree and type of workpieces at a high rate of speed. In [32], a dual-arm robot is used for surgical instrument sorting tasks using a coordinated control strategy, which combines a bilayer fuzzy hybrid force/position control method with a fuzzy control algorithm. The hybrid fuzzy control strategy is proposed for dual arm coordinated operations to dynamically adjust the motion parameters for efficient implementing instrument sorting tasks. Lin et al. [33] develop a fruit-sorting system using industrial robot and the three-dimensional (3D) visual perception. Robot can interact with human according to the real-time actual three-dimensional information and natural language interaction. Based on a 'rule-scene' matching and interaction algorithm, robot can sort the object automatically using the automatic programming and execution algorithm. Zhou et al. [34] develop an automatic sorting system for agricultural products using a 4-DOF robot arm and a monocular camera. An image processing method based on histogram correction was proposed for target classification. Target's position is calculated based on the pinhole imaging principle. The experiments conducted using tomatoes and oranges as the test objects have shown the reliability and high efficiency of the proposed system.

Robot is widely used in many assembly applications with the supporting of the vision system [35-39]. An automatic vision-guided system for a fruit picking and placing robot is developed by Dewi et al. [40]. A simple image processing algorithm based on BLOB analysis is presented to recognize the object by color and shape. The robot is equipped with a PI camera to capture the image of the fruit. The image is processed on the Raspberry Pi for detection processes. The experiment shows the effectiveness of the proposed method with the average time of picking and placing fruit is from 6.69s to 7.63s. Wei et al. [41] design a vision-guided manipulator to assemble workpieces without explicit human programming. A 3D camera is used to obtain the spatial positions of the workpiece dynamically during the demonstration, and then transform the positions to the coordinate system of the industrial manipulator. The inverse kinematic model of the industrial robot manipulator is applied to obtain the joint angles for each position of the workpiece. After that the robot manipulator is controlled to complete the task automatically based on the control information generated above. Instead of using 3D coordinates obtained from 3D camera, Jia et al. [42] use the features from the image acquired by a camera during the teaching process to generate the robot's trajectory. The desired trajectory is defined by using the homography instead of the coordinate of image feature points for more robust to image noise.

Many applications developed stereo vision systems to determine the position of objects for robot grasping [43-48]. Chen et al. [49] develop the picking robot system based on a Fuzzy Neural Network Sliding Mode Algorithms. Firstly, the kinematics and dynamics equations of the picking robot is established. Secondly, visual positioning is introduced to determine the position of the target point for an image-based visual control. Finally, the improved fuzzy neural network sliding mode control algorithm is proposed to carry out simulation analysis to improve the efficiency of the robot arm servo control. Cong et al. [50] develop a stereo vision system

to predict the 3D ball trajectory for the ping-pong robot. A multi-threshold segmentation algorithm and the triangulation algorithm are combined to compute the 3D coordinates of the ball. The aerodynamics model and rebound model are deployed to predict the flight trajectory of the ball. In [51], stereo vision-based system is used to extract 3-D coordinates between robot's tool center point (TCP) and the object of interest for multiagent robot application that can be used for tracking, tooling or handling operations. Cong and Hanh [52] use a combination of two visual servoing techniques in contour following task of unknown object. A camera is attached to the robot's end-effector to capture the image of the object. Shortest Path Visual Servoing is applied to regulate camera parallel to contour plane. The image-based visual servoing is applied for contour following task.

Recently, Deep Convolutional Neural Network (ConvNet or CNN), has been applied in many object recognition tasks which outstanding image recognition capabilities [53-56]. In [53], Li and Chang propose an innovative automation system for visual placement and precision positioning of the workpiece using a mobile manipulator. The system includes a binocular eye-in-hand system with two low-resolution CMOS cameras attached on the end effector of the mobile manipulator and a Convolutional Neural Network for detecting the relative position of the workpiece on the fixture. In order to estimating the grasping pose of SCARA robot, Wang et al. [54] use an end-to-end deep learning method on point clouds, PointNetRGPE. Point cloud data of objects is obtained from a binocular stereo vision system set directly above the objects.

III. DESCRIPTION AND PERFORMANCE OF ROBOTIC ARM SYSTEM

In this paper, a multi-robotic arm and conveyor belt system is developed for sorting objects. The robotic arm system consists of one master and three slave robots associated with three conveyor belts (Figure 1). The robot arms use parallelogram mechanisms to keep the end-effector always parallel to the horizontal during the robot motion. Each robotic arm is controlled by a robot controller based on a microcontroller. The joints of the robot arms are driven by 5.17:1 planetary gearbox stepper motor. The stepper motor can provide a maximum holding torque of 0.25Nm, resulting in a maximum robot's payload of 500g. In addition, the motors use a maximum of only 10W of power each, and three motors combined use a maximum of only 30W, resulting in significant energy savings. Table 1 shows the specifications of the robotic arms.

Objects in the workspace are captured by a stereo vision system consisting of two cameras. The images are processed by a master controller based on an embedded computer. A disparity map generated from the stereo vision system is used to detect and recognize objects. The depth of objects is also easy to determine from the disparity map, and then calculate the 3D coordinates of the objects. The position of the object is transmitted to the master robot using the Modbus RTU protocol through an RS485 serial interface. The master robot picks and puts the object on the right conveyor branch according to their shape. The three conveyors correspond to three different shapes of the object.

The conveyors move the objects to the position of the slave robot arm. The object is detected by a proximity sensor fixed on the conveyor. Based on the size attribute that the slave robot receives from the master, the object is picked and placed in the right position. The end-effector of the robot is attached to a vacuum suction cup to grip and move objects. Each slave robot arm is connected to the Arduino microcontroller. The Arduino board creates pulses and sends them to three A4998 drivers to driven stepper motors, reads the signal from the proximity sensor to control the motor conveyor by outputting a signal to a DC motor driver. In addition, the Arduino also outputs digital signals to control the solenoid valve, vacuum pump of the vacuum system for the vacuum suction cup. Since the valve and pump operate at 12V, relays whose coils are energized by the 5V signal from the Arduino are used to turn ON and OFF them. A UART to RS-485 converter module is used to allow the TTL interface of the microcontroller to be transferred to the RS485. Fig. 2 shows the block diagram of the robot controller.

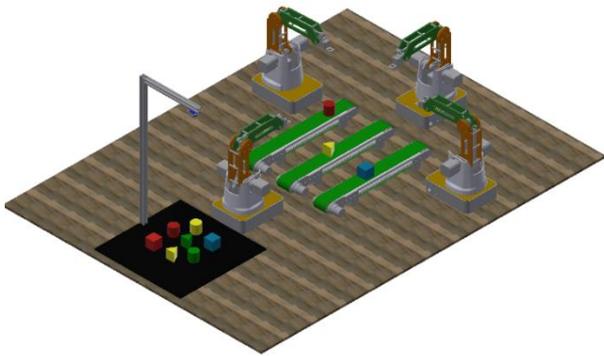


Fig. 1. Multi-robotic arm system

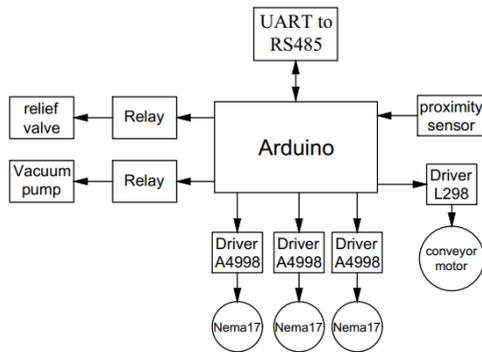


Fig. 2. Robot controller block diagram

IV. STEREO VISION SYSTEM

Stereo vision is a method to compute the depth of objects in order to construct three-dimensional model by using a pair of photographs of the same scene which are taken at different positions. Stereo vision has been widely used in many applications, such as cinema, video games, robot navigation, simultaneous localization and mapping (SLAM) as well as in many other aspects of production, security, defence, exploration. In a stereo vision system, a pair of images of an object is taken by a binocular camera consisting of two cameras as shown in Fig. 3. The object of interest is located in the real world at P, two origins of the left and right cameras are located at O_l and O_r , respectively. Assuming the two cameras are parallel, the right camera is a distance to the left

camera called baseline b . The projections of P on the left and right image planes have horizontal coordinates x_l and x_r , respectively. The disparity ($x_l - x_r$) can be determined by a simple expression:

$$d = x_l - x_r = \frac{bf}{Z} \tag{1}$$

where Z is the depth of the object, f is the focal length of two cameras. If a disparity map is determined, we can compute the depth Z of the object:

$$Z = \frac{bf}{d} \tag{2}$$

The challenging problem of establishing a disparity map is to find for each point in the left image, the corresponding point in the right one. Many stereo correspondence or stereo matching methods have been proposed, mainly in pursuit of real-time execution speeds, as well as decent accuracy. There are three common classes of techniques used for stereo matching: area-based [5], [6] feature-based [7], and phase-based [8]. Sum of Absolutely Difference (SAD) is the most favorable area-based technique in real-time stereo vision since it can be straightforwardly implemented in hardware.

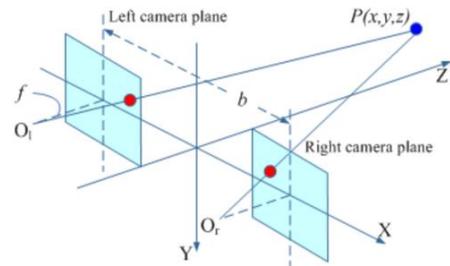


Fig. 3. Principle of stereo vision

Rebuilding the entire disparity map is computationally expensive. Instead, in this paper, we will find the coordinates of the centroids of the objects in the two images and use equation (2) to determine the Z depth value. After that, we can calculate 3D coordinates of the object by using the formula of the pinhole camera model:

$$\begin{aligned} X &= Z \frac{C_x - u_0}{f} \\ Y &= Z \frac{C_y - v_0}{f} \end{aligned} \tag{3}$$

where f is the focal length of cameras, (u_0, v_0) is the center of the image.

The 3D coordinates of the object's centroid in the camera frame are transformed to the robot frame using equation (11):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \tag{4}$$

where $[x, y, z]^T$ and $[X, Y, Z]^T$ are the coordinates of the object in the robot frame and the camera frame, respectively, R is the rotation matrix, t is the translation vector, representing the relationship between the two frames. Fig. 4 shows the flow chart to calculate the 3D coordinate of objects.

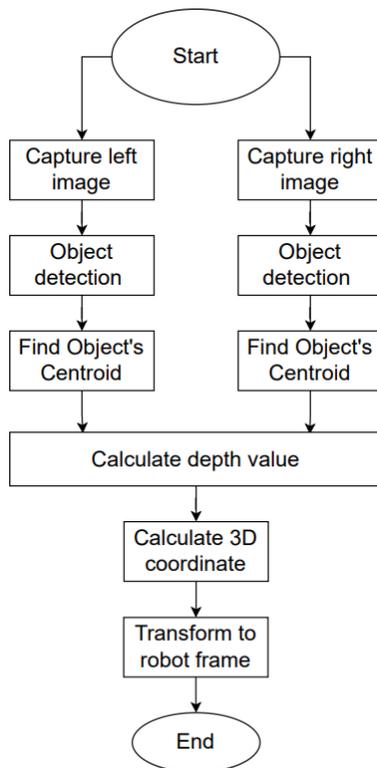


Fig. 4. Flow chart to calculate the 3D coordinate of objects

To extract objects from the background, the algorithm described in the previous studies [2] is applied. After extract the objects in images, we can compute image moments of objects and other features for classification and localization. The image moments are calculated according to the following formula:

$$M_{ij} = \sum_u \sum_v u^i v^j B(u, v) \quad (5)$$

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

From the image moment we can determine the area, A and centroid (C_u, C_v) of the objects.

$$A = M_{00} = \sum_u \sum_v I(u, v) \quad (6)$$

$$C_u = \frac{M_{10}}{M_{00}}$$

$$C_v = \frac{M_{01}}{M_{00}}$$

The area is used for size classification. To classify the shape of the object, compactness is used:

$$c = \frac{p^2}{A} \quad (7)$$

where c is the compactness, p is the perimeter and A is the area. The perimeter is calculated by summing all pixels on the contour of the object.

Some objects have overlapping compactness values in some cases. Therefore, to accurately identify the object, we use an additional attribute, called the fullness.

$$F = \frac{A_p}{A} \quad (8)$$

where F is the fullness, A_p is the area of a rectangle that approximates the object.

The threshold values for classification will be determined by experiment.

V. COMMUNICATION

In this project, robots are communicated through the Modbus RTU protocol that uses RS485 technology to wire Modbus Master and Slave devices. Fig. 5 shows the connection of the system. It is a 2 wire Modbus network corresponding to Half-Duplex communication. Converter modules are used to convert UART module to the RS485 communication standard. The master is a Raspberry Pi computer, slaves are Arduino boards used to control robot. Each slave must be connected to a common GND and wire to the master. Each device has 'A' and 'B' wires connected to form a half-duplex communication channel. A+ connects to A+ and B- connects to B-. Two wires A+ and B- are combined to create a signal. Logic value 1 is transmitted if line A is low and line B is high. Value 0 is transmitted if line A is high and line B is low. In this connection (half-duplex solution), the signal can travel only in one direction at once. Only one device can broadcast a signal at a time. The slaves only respond when a request message is received from the master.

Wiring devices in RS485 is easy. It has a multipoint topology, where one master is connected to multiple slaves. Each slave can be addressed with its own ID. We can connect a maximum of 247 slave devices (by 247 different ID-s) to a master device. The advantage of the RS485 connection is that signals can be transmitted faster and over greater distance than possible with a single wire RX/TX RS232 connection.

To initiate transactions (referred to as queries), the master sends a request message on the bus to a slave device or broadcasts to all the slave devices. The slave device takes action as per the request received and responds if required in the form of a "response message". Fig. 6 depicts a transaction of Modbus.

When transmitting a message, the entire message frame must be sent as a continuous stream of characters without inter-character hesitations. The message frame should be discarded by the receiver if the interval between two characters is more than 1.5-character times. Fig. 7 shows the Modbus RTU frame structure. Message frames are separated by a silent period of at least 3.5-character times. Each 8-bit byte is framed by 1 start bit, 8 data bits (least significant bit sent first), 0 or 1 parity bit, 1 or 2 stop bits. The different fields of Modbus frame structure are as follows:

Modbus Address: Modbus message starts with the 8-bit target slave address which is in the range of 0 – 247 decimal. Address 0 is used as broadcast address. The master places the slave address in the address field to specify the slave for receiving the request message. When the slave responds, it places its own address in the response address field to tell the master where the response message came from.

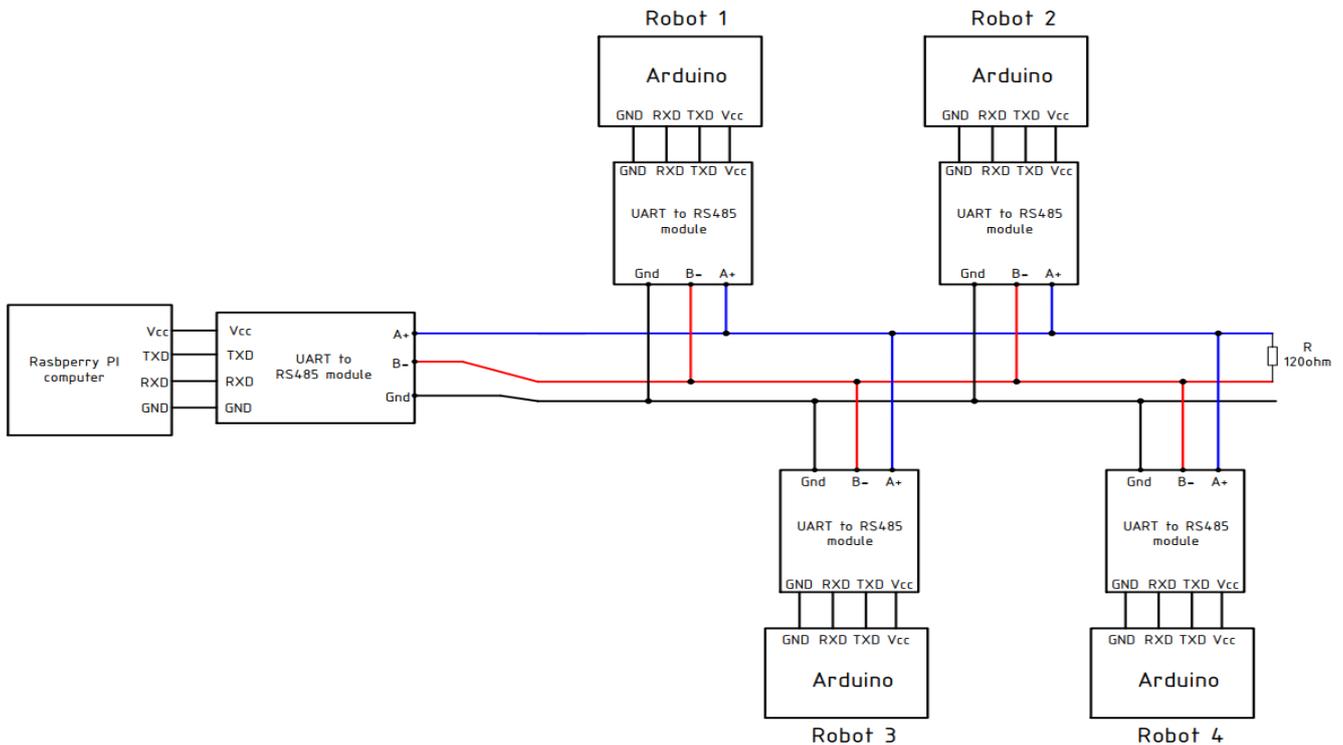


Fig. 5. 2-wire (half-duplex) connections

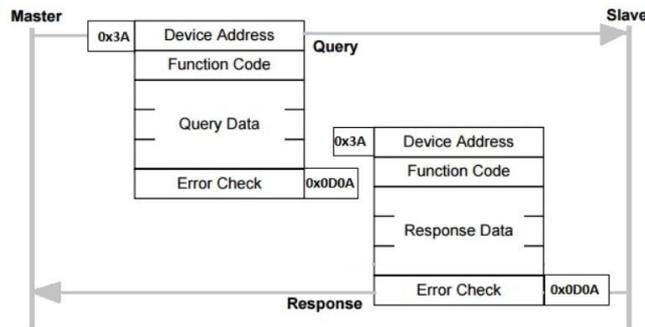


Fig. 6. Modbus message frames

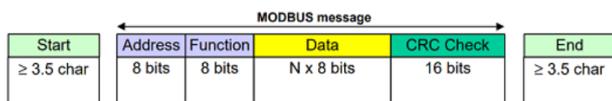


Fig. 7. Modbus RTU message frame structure

Function code: describe the kind of action that the master performs on the slaves. These actions can be "read" or "write" data on registers. Table 1 describes some common codes in Modbus. Discrete Inputs and Coils are 1-bit registers, Input Registers and Holding registers are 16-bit register.

TABLE I. TABLE TYPE STYLES

Function code	Function name
01(0x01)	Read Coil Status
02(0x02)	Read Discrete Inputs
03(0x03)	Read Multiple Holding Registers
04(0x04)	Read Input Registers
05(0x05)	Write Multiple Coils
06(0x06)	Write Single Holding Register
15(0x0F)	Write Multiple Coils
16(0x10)	Write Multiple Holding Registers

Data Field: contains request and response parameters. The maximum length of the data is 252 bytes.

CRC check Field: contains a 16-bit error-detecting code to detect accidental changes to raw data. The CRC field is calculated by the master and appended to last in the message where the low-order byte is appended first, followed by the high-order byte. The slave device calculates a CRC during receipt of the message and compares the calculated value to the CRC in the received message. If the two values do not match, the message should be ignored. Fig. 8 shows the code to create CRC on C/C++.

```

// Compute the MODBUS RTU CRC
unsigned int ModRTU_CRC(unsigned char* buf, int len)
{
    int i,j;
    unsigned int crc, temp;
    crc = 0xFFFF;

    for (i = 0; i < len; i++) {
        crc ^= buf[i];

        for (j = 0; j < 8; j++) {
            crc >>= 1;
            if ((crc & 0x0001) != 0) {
                crc ^= 0xA001;
            }
        }
    }

    temp = crc >> 8;
    crc = (crc << 8) | temp;
    return crc;
}
    
```

Fig. 8. C/C++ code for calculate CRC

The type of object and robot's joint angles are sent by the master to robots on the Modbus network. The robot's joint

angles include three angle values of three rotation axes. Therefore, to transmit joint angles, function code 16 is used. Table 2 and Table 3 show the Data Field in the request and response messages of function code 16.

TABLE II. TABLE TYPE STYLES

Starting Address	2 bytes	0x0000 to 0xFFFF
Quantity of Registers	2 bytes	0x0001 to 0x007B
Byte Count	1 byte	2 x N
Registers Value	N x 2 Bytes	value

TABLE III. TABLE TYPE STYLES

Starting Address	2 bytes	0x0000 to 0xFFFF
Quantity of Registers	2 bytes	0x0001 to 0x007B

In detail, for sending the coordinates of the object, the request message is:

Addr 0x10 0x0001 0x0003 0x06 T1 T2 T3 CRC

where

- **Addr:** The Slave Address
- **0x10:** The Function Code 16 (Write Multiple Holding Registers)
- **0x0001:** The Data Address of the first register
- **0x0003:** The number of registers to write.
- **0x06:** The number of data bytes to follow (3 registers x 2 bytes each = 6 bytes).
- **T1:** The rotation angle of joint 1
- **T2:** The rotation angle of joint 2
- **T3:** The rotation angle of joint 3
- **CRC:** The CRC (Cyclic Redundancy Check) for error checking.

And the response message from the slave is:

Addr 0x10 0x0001 0x0003 0x06 CRC

where:

- **Addr:** The Slave Address
- **0x10:** The Function Code 16 (Write Multiple Holding Registers)
- **0x0001:** The Data Address of the first register
- **0x0003:** The number of registers to write.
- **CRC:** The CRC (Cyclic Redundancy Check) for error checking.

T1, T2 and T3 are 16-bit integer values. The actual rotation angle values are not integers. However, they are rounded to 2 decimal places and multiplied by 100 to become an integer before being sent. The rotation angles have values from -180 degrees to 180 degrees. So, a 16-bit integer is enough to represent them.

The type of objects is encoded as an ASCII character. Therefore, to transmit type of objects, function code 06 is used. Table 4 shows the Data Field in the request and response messages of function code 06

TABLE IV. TABLE TYPE STYLES

Starting Address	2 bytes	0x0000 to 0xFFFF
Registers Value	2 Bytes	0x0000 to 0xFFFF

In detail, the request and respond messages for sending the type of objects are:

Addr 0x06 0x0004 P CRC

where:

- **Addr:** The Slave Address
- **0x06:** The Function Code 16 (Write Single Holding Registers)
- **0x0004:** The address of the register
- **P:** type of product (8-bit value)
- **CRC:** The CRC (Cyclic Redundancy Check) for error checking.

VI. RESULTS AND DISCUSSION

The stereo vision system uses two cameras with resolution of 480x640 pixels. Two cameras are mounted parallel to each other by using a fixture (see Fig. 9). The distance between two camera is 60mm. A sample images of objects taken by two camera is shown in Fig. 10.



Fig. 9. Two cameras of the stereo vision system

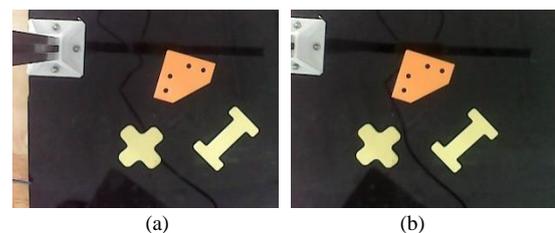


Fig. 10. Sample Images. (a) left image (b) right image

The images are processed to create binary images by using our algorithm presented in [2]. Then, the thresholds of the compactness and fullness are applied to classify objects. By experiment, we determine ranges of two thresholds to classify three objects as follows:

Object 1:

$$17 \leq c \leq 19$$

$$1.6 \leq F \leq 1.7$$

Object 2:

$$23.5 \leq c \leq 25.5$$

$$1.4 \leq F \leq 1.6$$

Object 3:

$$36 \leq c \leq 38$$

$$3.1 \leq F \leq 3.3$$

After classifying objects, we find the centroids of objects in the image by using Equation (6). These coordinates are used to calculate the 3D coordinates of objects by using Equation (2) and Equation (3). The experiment results show that the estimate method can determine accurately the coordinate of the object.

From the analysis of the Modbus RTU protocol using RS485 technology, we can write code to implement on Raspberry Pi and Arduino. The function code to send Angles is shown in Fig. 11. This function is used to send a rotation angle from the master to the slave robot controller. It has four inputs, the first is the address of the slave, and the other three values are the rotation angles of the three joints. This function is written in Python language and runs on Raspberry Pi.

```
#Function: sendAngle
#
# Send joint angles to the robot
#
# Addr: Slave address
# T1: rotation angle of Joint 1
# T2: rotation angle of Joint 2
# T3: rotation angle of Joint 3
#
# Return: None

def sendAngle(Addr, T1, T2, T3):
    buf= array.array('B', [Addr,16,0,1,0,3,6])
    buf.append(T1 >> 8)
    buf.append(T1 & 0xFF)
    buf.append(T2 >> 8)
    buf.append(T2 & 0xFF)
    buf.append(T3 >> 8)
    buf.append(T3 & 0xFF)
    crc = ModRTU_CRC(buf, 13)
    buf.append(crc >> 8)
    buf.append(crc & 0xFF)
    for data in buf:
        ser.write(data)
```

Fig. 11. Function to send angles

The function code to send type of object is shown in Fig. 12. This function is used to send type of objects from the master to the slave robot controller. It has two inputs, the first is the address of the slave, and another value is a symbol describing the object. This function is also written in Python language and runs on Raspberry Pi.

```
#Function: sendType
#
# Send type of the product the robot
#
# Addr: Slave address
# P: type of product
#
# Return: None

def sendAngle(Addr, P):
    buf= array.array('B', [Addr,6,0,4,P])
    crc = ModRTU_CRC(buf, 5)
    buf.append(crc >> 8)
    buf.append(crc & 0xFF)
    for data in buf:
        ser.write(data)
```

Fig. 12. Function to send type of objects

The functions are written by C/C++ language and implement on Arduino to read message from the master is shown in Fig. 13 and Fig. 14. The function for reading angle is shown in Fig. 14. It is used to read three angle values for the joints of the robot from the master. The function for reading the type of objects is shown in Figure 15. Practical tests show that these functions work well to exchange information between the master and slaves.

```
/*
 *Function: readAngle
 *
 *read rotation angle
 *
 *Addr: Slave address
 *
 *Return: 1 if successful transmission
 *       0 if fail
 */
char readAngle(unsigned char Addr){
    int i = 7;
    char buf[15] = {Addr,0x10,0x00,0x01,0x00,0x03,0x06}
    while(i<15){
        if(Serial.available()){
            buf[i] = Serial.read();
            i++;
        }
    }
    unsigned int crc_cal = ModRTU_CRC(buf, 13);
    unsigned int crc_rec = (buf[13]<< 8) | buf[14];
    if(crc_cal == crc_rec){
        unsigned int a1 = (buf[7] << 8) | buf[8];
        unsigned int a2 = (buf[9] << 8) | buf[10];
        unsigned int a3 = (buf[11] << 8) | buf[12];
        angle1 = a1/100.0;
        angle2 = a2/100.0;
        angle3 = a3/100.0;
        return 1;
    }
    else
        return 0;
}
```

Fig. 13. Function to read angles

```
/*
 *Function: readType
 *
 *read type of the object
 *Addr: Slave address
 *
 *Return: 1 if successful transmission
 *       0 if fail
 */
char readType(unsigned char Addr){
    int i = 4;
    char buf[7] = {Addr,0x06,0x00,0x04}
    while(i<7){
        if(Serial.available()){
            buf[i] = Serial.read();
            i++;
        }
    }
    unsigned int crc_cal = ModRTU_CRC(buf, 5);
    unsigned int crc_rec = (buf[5]<< 8) | buf[6];
    if(crc_cal == crc_rec){
        type = buf[4];
        return 1;
    }
    else
        return 0;
}
```

Fig. 14. Function to read type of objects

VII. CONCLUSION

This paper has developed a multi-robot arm system to sort object based on a stereo vision system. The master robot and slave robots communicate data by using the Modbus RTU protocol through an RS485 serial interface. Through Modbus protocol analysis, functions for sending and receiving data are written in Python and C/C++ languages for the master and the slaves, respectively. The experiment results show that the robots can exchange data and operation on demand.

The stereo vision system is built to determine the 3D coordinates of the object. Instead of rebuilding the entire disparity map, which is computationally expensive, the centroids of the objects in the two images are calculated to determine the depth value. After that, we can calculate the 3D coordinates of the object by using the formula of the pinhole camera model.

ACKNOWLEDGMENT

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

REFERENCES

- [1] R. Song, F. Li, T. Fu and J. Zhao, "A Robotic Automatic Assembly System Based on Vision," *Applied Sciences*, vol. 10, no. 3, 2020, <https://doi.org/10.3390/app10031157>.
- [2] V. D. Cong, "Industrial robot arm controller based on programmable system-on-chip device," *FME Transactions*, vol. 49, no.4, pp. 1025-1034, 2021, <https://doi.org/10.5937/fme2104025C>.
- [3] M. Z. U. Rahman, M. Usman, A. Farea, N. Ahmad, I. Mahmood and M. Imran, "Vision-Based Mobile Manipulator for Handling and Transportation of Supermarket Products," *Mathematical Problems in Engineering*, vol. 7, 2022, <https://doi.org/10.1155/2022/3883845>.
- [4] F. Li, Q. Jiang, W. Quan, S. Cai, R. Song and Y. Li, "Manipulation Skill Acquisition for Robotic Assembly Based on Multi-Modal Information Description," *IEEE Access*, vol. 8, pp. 6282 – 6294, 2019, <https://doi.org/10.1109/ACCESS.2019.2934174>.
- [5] P. Guo, Z. Zhang, Y. Liu, Y. Liu, D. Zhu and C. Shao, "A Skill Programming Method Based on Assembly Motion Primitive for Modular Assembly System," *IEEE Access*, vol. 9, pp. 101369-101380, 2021, <https://doi.org/10.1109/ACCESS.2021.3080838>.
- [6] V. D. Cong, L.D. Hanh, L.H. Phuong, D.A. Duy, "Design and development of robot arm system for classification and sorting using machine vision," *FME Transactions*, vol. 50, no.1, 2022, <https://doi.org/10.5937/fme2201181C>.
- [7] T. Jiang, H. Cui, X. Cheng, W. Tian, "A Measurement Method for Robot Peg-in-Hole Preamplification Based on Combined Two-Level Visual Sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-12, 2021, <https://doi.org/10.1109/TIM.2020.3026802>.
- [8] Q. Zhao, X. Li, J. Lu and J. Yi, "Monocular vision-based parameter estimation for mobile robotic painting," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 10, pp. 3589-3599, Oct. 2019, <https://doi.org/10.1109/TIM.2018.2878427>.
- [9] F. Li, Q. Jiang, W. Quan, S. Cai, R. Song and Y. Li, "Manipulation skill acquisition for robotic assembly based on multi-modal information description," *IEEE Access*, vol. 8, pp. 6282-6294, 2020, <https://doi.org/10.1109/ACCESS.2019.2934174>.
- [10] L. D. Hanh and V.D. Cong, "Implement contour following task of objects with unknown geometric models by using combination of two visual servoing techniques," *International Journal of Computational Vision and Robotics*, vol. 12, no. 5, pp. 464-486, 2022, <https://doi.org/10.1504/IJCVR.2022.10048788>.
- [11] C. Cai, R. Qiao, H. Meng and F. Wang, "A novel measurement system based on binocular fisheye vision and its application in dynamic environment," *IEEE Access*, vol. 7, pp. 156443-156451, 2019, <https://doi.org/10.1109/ACCESS.2019.2949172>.
- [12] O. I. Abdullah, W.T. Abbood and H.K. Hussein, "Development of automated liquid filling system based on the interactive design approach," *FME Transactions*, vol. 48, no. 4, pp. 938-945, 2020, <https://doi.org/10.5937/fme2004938A>.
- [13] L. Caruana and E. Francalanza, "Safety 4.0 for Collaborative Robotics in the Factories of the Future," *FME Transactions*, vol. 49, no. 4, pp. 842-850, 2021.
- [14] V. D. Cong, L.D. Hanh, "A new decoupled control law for image-based visual servoing control of robot manipulators," *International Journal of Intelligent Robotics and Applications*, vol. 6, pp. 576-585, 2022, <https://doi.org/10.1007/s41315-022-00223-5>.
- [15] J. Wang, C. Hu, Y. Wang, Y. Zhu, "Dynamics Learning With Object-Centric Interaction Networks for Robot Manipulation," *IEEE Access*, vol. 9, pp. 68277-68288, 2021, <https://doi.org/10.1109/ACCESS.2021.3077117>.
- [16] C. Hu, T. Ou, Y. Zhu and L. Zhu, "GRU-type LARC strategy for precision motion control with accurate tracking error prediction," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 1, pp. 812-820, Jan. 2021.
- [17] B. Ichter and M. Pavone, "Robot motion planning in learned latent spaces," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2407-2414, Jul. 2019.
- [18] X. Li, J. Xiao, W. Zhao, H. Liu, G. Wang, "Multiple peg-in-hole compliant assembly based on a learning-accelerated deep deterministic policy gradient strategy," *Industrial Robot: the international journal of robotics research and application*, vol. 49, no. 1, pp. 54, 2022.
- [19] C. Weckenborg, K. Kieckhäfer, C. Müller, C. et al., "Balancing of assembly lines with collaborative robots," *Business Research*, vol. 13, pp. 93-132, 2020, <https://doi.org/10.1007/s40685-019-0101-y>
- [20] F. Zha, Y. Fu, P. Wang, W. Guo, M. Li, X. Wang, H. Cai, "Semantic 3D Reconstruction for Robotic Manipulators with an Eye-In-Hand Vision System," *Applied Sciences*, vol. 10, 2020, <https://doi.org/10.3390/app10031183>.
- [21] M. Intisar, M.M. Khan, M.R. Islam and M. Masud, "Computer Vision Based Robotic Arm Controlled Using Interactive GUI," *Intelligent Automation and Soft Computing*, vol. 27, no. 2, pp. 533-550, 2021, <https://doi.org/10.32604/iasc.2021.015482>.
- [22] H. Wei, X.X. Chen and X.Y. Miao, "Vision-guided fine-operation of robot and its application in eight-puzzle game," *International Journal of Intelligent Robotics and Applications*, vol. 5, no.3, pp. 576-589, 2021, <https://doi.org/10.1007/s41315-021-00186-z>.
- [23] A. Zakhama, L. Charrabi and K. Jelassi, "Intelligent Selective Compliance Articulated Robot Arm robot with object recognition in a multi-agent manufacturing system," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, 2019, <https://doi.org/10.1177/1729881419841145>.
- [24] H. M. Ali, Y. Hashim and G. A. Al-Sakkal, "Design and implementation of Arduino based robotic arm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 1411-1418, 2021, <https://doi.org/10.11591/ijece.v12i2>.
- [25] F. Fang, M. Shi, K. Qian, et al., "A human-aware navigation method for social robot based on multi-layer cost map," *International Journal of Intelligent Robotics and Applications*, vol. 4, pp. 308-318, 2020, <https://doi.org/10.1007/s41315-020-00125-4>.
- [26] A. H. Wei, B.Y. Chen, "Robotic object recognition and grasping with a natural background," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, pp. 1-17, 2020, <https://doi.org/10.1177/1729881420921102>.
- [27] M. M. Babr, M. Faghihabdolahi, D. Ristić-Durrant and K. Michels, "Deep Learning-Based Occlusion Handling of Overlapped Plants for Robotic Grasping," *Applied Sciences*, vol. 12, no. 7, 2022, <https://doi.org/10.3390/app12073655>.
- [28] A. Atefi, Y. Ge, S. Pitla and J. Schnable, "Robotic Detection and Grasp of Maize and Sorghum: Stem Measurement with Contact," *Robotics*, vol. 9, no. 3, 2020, <https://doi.org/10.3390/robotics9030058>.
- [29] M. Tahmasebi, M. Gohari and A. Emami, "An Autonomous Pesticide Sprayer Robot with a Color-based Vision System," *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 115-123, 2022, <https://doi.org/10.31763/ijrcs.v2i1.480>.

- [30] M. N. Sofu, O. Er, M.C. Kayacan and B. Cetisli, "Design of an automatic apple sorting system using machine vision," *Computers and Electronics in Agriculture*, vol. 127, pp. 395-405, 2016.
- [31] X. Wu, X. Ling and J. Liu, "Location recognition algorithm for vision-based industrial sorting robot via deep learning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 07, Jun. 2019.
- [32] Q. Wu, M. Li, X. Qi, Y. Hu, B. Li, and J. Zhang, "Coordinated control of a dual-arm robot for surgical instrument sorting tasks," *Robotics and Autonomous Systems*, vol. 112, pp. 1-12, Feb. 2019.
- [33] Y. Lin, H. Zhou, M. Chen and H. Min, "Automatic sorting system for industrial robot with 3D visual perception and natural language interaction," *Measurement and Control*, vol. 52, no. 1, pp. 100-115, Jan. 2019.
- [34] K. Zhou, Z. Meng, M. He, Z. Hou and T. Li, "Design and Test of a Sorting Device Based on Machine Vision," *IEEE Access*, vol. 8, pp. 27178-27187, 2020.
- [35] R. Song, F. Li, T. Fu and J. Zhao, "A Robotic Automatic Assembly System Based on Vision," *Applied Sciences*, vol. 10, no. 3, 2020, <https://doi.org/10.3390/app10031157>.
- [36] W. Wan, F. Lu, Z. Wu and K. Harada, "Teaching robots to do object assembly using multi-modal 3D vision," *Neurocomputing*, vol. 25, no. 9, pp. 85-93, 2017.
- [37] H. Chen, J. Xu, B. Zhang, T.A. Fuhlbrigge, "Improved parameter optimization method for complex assembly process in robotic manufacturing," *Industrial Robot*, vol. 44, no. 1, pp. 21-27, 2017, <https://doi.org/10.1108/IR-03-2016-0098>.
- [38] L. Yang, M. Chong, C. Bai and J. Li, "A multi-workpieces recognition algorithm based on shape-SVM learning model," *Journal of Physics Conference Series*, vol. 1087, no. 2, 2018, <https://doi.org/10.1088/1742-6596/1087/2/022025>.
- [39] L. Roveda, "Adaptive Interaction Controller for Compliant Robot Base Applications," *IEEE Access*, vol. 7, pp. 6553-6561, 2019.
- [40] T. Dewi, Z. Mulya, P. Risma and Y. Oktarina, "BLOB analysis of an automatic vision guided system for a fruit picking and placing robot," *International Journal of Computational Vision and Robotics*, vol. 11, no. 3, pp.315-327, 2021.
- [41] Q. Wei, C. Yang, W. Fan and Y. Zhao, "Design of demonstration-driven assembling manipulator," *Applied Sciences*, vol. 8, no. 5, pp. 797, May 2018.
- [42] B. Jia, S. Liu and Y. Liu, "Visual trajectory tracking of industrial manipulator with iterative learning control," *Industrial Robot: An International Journal*, vol. 42, no. 1, pp. 54-63, 2015.
- [43] Y.C. Du, T. Taryudi, C.T. Tsai, et al., "Eye-to-hand robotic tracking and grabbing based on binocular vision," *Microsystem Technologies*, vol. 27, no. 4, pp. 1699-1710, 2021, <https://doi.org/10.1007/s00542-019-04475-3>.
- [44] Taryudi and M.S.Wang, "Eye to hand calibration using ANFIS for stereo vision-based object manipulation system," *Microsystem Technologies*, vol. 24, no. 1, pp. 305-317, 2018, <https://doi.org/10.1007/s00542-017-3315-y>.
- [45] Y.C. Du, M. Muslikhin, T.H. Hsieh and M.S. Wang, "Stereo Vision-Based Object Recognition and Manipulation by Regions with Convolutional Neural Network," *Electronics*, vol. 9, no. 2, 2020, <https://doi.org/10.3390/electronics9020210>.
- [46] R. Holubek and M. Vagaš, "Center of Gravity Coordinates Estimation Based on an Overall Brightness Average Determined from the 3D Vision System," *Applied Sciences*, vol. 12, no. 1, 2022, <https://doi.org/10.3390/app12010286>.
- [47] C.A. Aguilera, C. Aguilera, C.A. Navarro and A.D. Sappa, "Fast CNN Stereo Depth Estimation through Embedded GPU Devices," *Sensors*, vol. 20, no. 11, 2020, <https://doi.org/10.3390/s20113249>.
- [48] A. Mohamed, P. Culverhouse, A. Cangelosi, et al., "Active stereo platform: online epipolar geometry update," *EURASIP Journal on Image and Video Processing*, vol. 54, 2018, <https://doi.org/10.1186/s13640-018-0292-8>.
- [49] W. Chen, T. Xu, J. Liu, M. Wang and D. Zhao, "Picking robot visual servo control based on modified fuzzy neural network sliding mode algorithms," *Electronics*, vol. 8, no. 6, pp. 605, May 2019.
- [50] V.D. Cong, L.D. Hanh, L.H. Phuong, "Real-time Measurement and Prediction of Ball Trajectory for Ping-pong Robot," 2020 5th International Conference on Green Technology and Sustainable Development (GTSD), pp. 9-14, 2020.
- [51] F. Šuligoj, B. Šekoranja, M. Švaco, B. Jerbić, "Object tracking with a multiagent robot system and a stereo vision camera," *Procedia Engineering*, vol. 69, pp. 968-973, 2014.
- [52] V.D. Cong, L.D. Hanh, "Combination of two visual servoing techniques in contour following task," 2021 International Conference on System Science and Engineering (ICSSE), pp. 382-386, 2021.
- [53] C.H.G. Li, Y.M. Chang, "Automated visual positioning and precision placement of a workpiece using deep learning," *International Journal of Advanced Manufacturing*, vol. 104, pp. 4527-4538, 2019.
- [54] Z. Wang, Y. Xu, Q.He et al., "Grasping pose estimation for SCARA robot based on deep learning of point cloud," *International Journal of Advanced Manufacturing*, vol. 108, pp. 1217-1231, 2020.
- [55] D. Zhao, F. Sun, Z. Wang et al., "A novel accurate positioning method for object pose estimation in robotic manipulation based on vision and tactile sensors," *International Journal of Advanced Manufacturing*, vol. 116, pp. 2999-3010, 2021.
- [56] Z. Wang, Y. Xu, G. Xu, J. Fu, J. Yu and T. Gu, "Simulation and deep learning on point clouds for robot grasping," *Assembly Automation*, vol. 41, no. 2, pp. 237-250, 2021.
- [57] S. Kumra, S. Joshi and F. Sahin, "GR-ConvNet v2: A Real-Time Multi-Grasp Detection Network for Robotic Grasping," *Sensors*, vol.22, no. 16, 2022, <https://doi.org/10.3390/s22166208>.
- [58] T. Zhang, C. Zhang, T. Hu, "A robotic grasp detection method based on auto-annotated dataset in disordered manufacturing scenarios," *Robotics and Computer-Integrated Manufacturing*, vol. 76, 2022, <https://doi.org/10.1016/j.rcim.2022.102329>.
- [59] K. Kleeberger, R. Bormann, W. Kraus, et al., "A Survey on Learning-Based Robotic Grasping," *Current Robotics Reports*, vol. 1, pp. 239-249, 2020, <https://doi.org/10.1007/s43154-020-00021-6>.
- [60] M. Hegedus, K. Gupta and M. Mehrandehz, "Efficiently finding poses for multiple grasp types with partial point clouds by uncoupling grasp shape and scale," *Autonomous Robots*, vol. 46, no. 4, pp. 749-767, 2022, <https://doi.org/10.1007/s10514-022-10049-6>.