GRASPING AND REPOSITIONING OBJECTS USING INVERSE KINEMATIC METHOD FOR ARM ROBOT BASED ON PIXEL POSITION REGRESSION

S. Sendari[1], Y.R. Wahyudi[1, 2], I.A.E. Zaeni[1], A.N. Handayani[1], M. Muladi[1], N. Wicaksono[1], M.A. Fatwaddin[1,2] and H. Lin[2]

[1]Faculty of Engineering,
Universitas Negeri Malang, Jalan Semarang No. 5, Malang, 65145, Indonesia.

[2] Graduate Institute of Automation,
National Taipei University of Technology,
Rm 801A, Integrated Technology Complex, I, Sec. 3,
Chung-Hsiao E Road, Taipei 106, Taiwan

Corresponding Author's Email: [1]siti.sendari.ft@um.ac.id

Article History: Received 10 February 2022; Revised 22 October 2022; Accepted 8 December 2022

ABSTRACT: This paper presented a method of recognizing objects' positions using a vision sensor to identify the grasping point of the arm robot. The arm robot used an embedded color vision Pixy CMUcam5 camera (Pixy camera) to capture and process the colored object image. Pixy camera is low in cost and easily programmed with high-speed FPS in real-time data processing. However, it still has the fisheye effect problem when recognizing the object's position. Thus, the pixel position regression algorithm was chosen to transform the colored objects' positions in the real coordinates. This method was implemented in the arm robot for grasping, picking, and repositioning tasks. The transformation results were employed to calculate the kinematics of the arm robot's joints. The experimental results showed that the Pixy camera could identify objects in the real world and transform the objects' positions. The error value at the real-world position was no more than 1.67 cm, and the experiment showed that the success rate of doing the task was up to 80%.

KEYWORDS: *Arm Robot, Embedded Color Vision, Object Recognition, Pixel Position Regression, Repositioning*

1.0 INTRODUCTION

Advanced robot technologies have been widely used in various fields to assist people in carrying goods and doing repetitive tasks such as picking and placing goods that require a safe and high accuracy level [1]. Arm-robot is commonly used since it resembles the human's arm. Recently, various robotic arm applications have been developed, such as the arm robot for the sorting system [2] and the warehouse automation system [3] to pick and place various goods on the shelves. Accurate object position estimation is crucial to pick an object successfully [4].

Object repositioning is a primary task that the arm robot should complete. It is used in many applications, such as warehouse automation systems, to grasp and pick up objects and put them in the bin [5]. Thus, the arm robot should be capable of identifying the targeted object, including the type, position, and size, and identifying other objects, such as bins, shelves, and other features that affect its movements [3]. Recently, a camera sensor has been widely used as vision-based object recognition [6, providing a powerful tool to capture image data. Therefore, the arm robot combined with a vision sensor can detect the object precisely and recognize its position.

Several approaches have discussed the method to solve the problem of detecting the object's position based on vision data [6, 7]. The method uses a learning process combined with an artificial intelligence system [1], which is widely used in classification systems [8, 9]. The learning algorithm for vision-based detection could extract useful information to determine its movements. However, the learning process computation is challenging because it requires complex algorithms and high-quality hardware. Implementing the embedded color vision can drastically reduce the time and cost instead of computing the image processing. The embedded color vision, such as the Pixy CMUcam5 camera (Pixy camera), is low in cost and easily programmed with a high-speed Frame per Second (FPS) in real-time data processing [11]. Pixy camera is also flexible and can be connected directly to a microcontroller-based system, such as Arduino and Raspberry Pi [12]. Furthermore, the Pixy camera can track the robot's movements using a color combination (CC) and pixel position [13].

This research aimed to solve the problem of grasping and positioning objects using an arm robot to determine the robot movements combined with the inverse kinematic method [13, 14, 15]. Regardless of the advantage, the Pixy camera still has the fisheye effect problem when used to recognized the object's position. Thus, the pixel position regression method was applied to transform the pixel coordinate into the real-world position. Therefore, this paper studied this method's effectiveness in implementing the Pixy camera.

2.0 METHODOLOGY

2.1 System Design

Figure 1(a) shows the arm robot system's experimental environment. The area is 600 × 300 mm, with the camera sensor installed 40 cm above the field. The camera is faced down to capture the reached area by the arm robot, as shown in Figure 1 (a). The system is divided into three parts, as shown in Figure 1(b); vision sensor (Pixy camera), main controller (Arduino) and arm robot actuator. The working space of the arm robot is presented in Figure 1(c). The arm robot worked in a 180 degrees range with a cubed object of 30 × 30 × 30 mm.

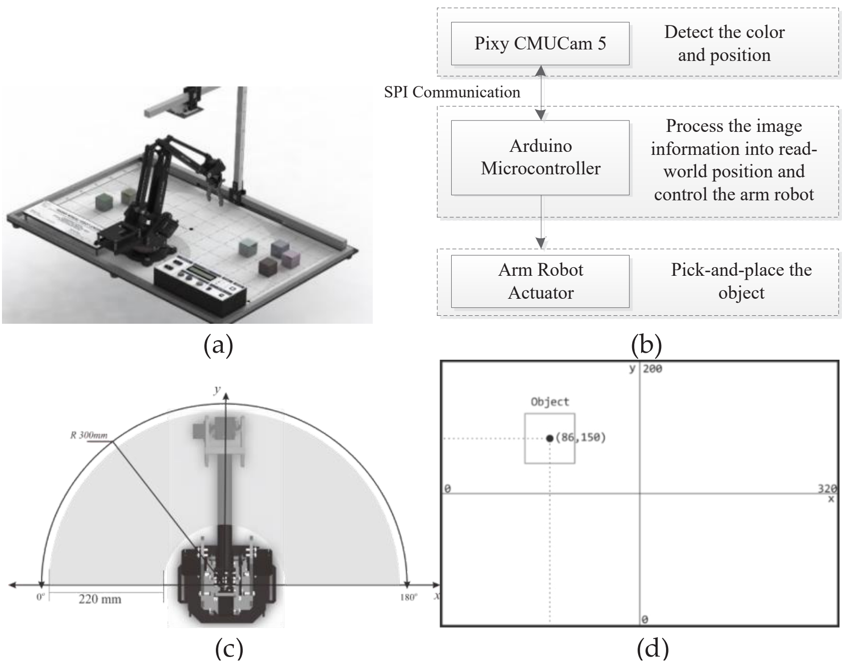


Figure 1: (a) Arm Robot System Design, (b) System Diagram, (c) Working Space Arm Robot, and (d) Illustration of Object Position Detection Robot

The camera was calibrated using a software interface provided by the Pixy camera module and trained with four colored objects: red, green, blue, and yellow. When the camera detected the objects, it generated information about the object's color id, pixel position, and size (width and height). Then, the camera transmitted the data to the Arduino microcontroller through SPI communication.

The unique point of this proposed method is that the object position information is determined according to real-world coordinates. Since the Pixy camera has a fisheye effect problem, the algorithm of pixel position regression was chosen to transform the colored objects' position in real-world coordinates. The objective was for the arm robot to determine the movements of the end-effector to grasp the objects and relocate them to new positions.

2.2 Embedded Color Vision

The Pixy camera utilizes a filtering algorithm to determine the objects based on the color and saturation data [11]. The Pixy camera was chosen since it is low in budget and can recognize colored objects fast. The captured image has a resolution of up to 300 × 200 and up to 50 fps frame rate [10]. The camera can record and memorize seven colors represented as the object's ID. Pixy camera provided data in the center position (x, y) and the detected object size in the covered area. The center position was useful to determine the angle values of the arm robot's joints. Regarding these specifications, the object positions could be interpreted and submitted easily to the external microcontroller ATMega328-based, i.e., Arduino Uno.

2.3 Arm Robot Design

The arm robot was developed using 3-DOF, which consists of the base, shoulder, and elbow. A gripper was installed in the end-effector of the arm robot for picking up the objects, while the wrist joint was used to turn the end-effector. The arm robot could move in the 3D working area (x, y, z coordinates). The base joint could turn horizontally at 180 degrees so that the robot could turn left or right (x-axis). The shoulder and elbow joints were used to turn vertically, so it could go up or down (z-axis) and move forward or backward (y-axis). The actuators of the base, shoulder, and elbow joints used the Robostar SRS-3216HTG servo, while the wrist used the Mg90 micro servo.

The maximum for the gripper to open was up to 45 mm so that it could grasp the object of 40 × 40 × 40 mm dimensions. The robot consisted of two main links actuated by the shoulder and elbow joints. The shoulder link dimension was 148 mm, and the elbow link was 160 mm. The arm working range was 300 mm – 80 mm, and the robot could rotate up to 180 degrees. The working space of the arm robot is shown in Figure 1(c).

2.4 Pixel Position Regression

The object's ID and position data were critical information for the arm robot to grasp and reposition the objects. The object's ID was obtained through the sampling and training process in the pixy camera module, and the real-world position was predicted by transforming the pixel position information. Then, the object's ID and pixel position data provided by the Pixy camera module were transmitted to the Arduino-Uno microcontroller through SPI communication.

Pixy camera provided information about the objects' position represented by the pixel position (P_p) marked by X_p and Y_p. The maximum of X_p was equal to the horizontal resolution of the Pixy image, so the range of X_p was 0–320. Furthermore, the maximum of Y_p was equal to the image's vertical resolution, which meant the Yp range was 0–200.

The pixel position regression method was implemented to transform the pixel position (P_p) information to the real-world position (P_r). This method used pixel position (P_p) as the independent variable and real-world position (P_r) as the dependent variable. This research used pixel position regression to fit the straight line through the data that predicted P_r (real position) based on P_p (pixel position). This model is expressed as:

$$P_r = a + bP_p \tag{1}$$

The regression parameters are the intercept of the positions, and b is the gradient [17]. The parameters of a and b can be calculated using the equation as follow to obtain the equation that describes the relationship between P_p and P_r:

$$b = \frac{n\sum_{i=1}^{n} P_{r_i} P_{p_i} - \left(\sum_{i=1}^{n} P_{r_i}\right)\left(\sum_{i=1}^{n} P_{p_i}\right)}{n\sum_{i=1}^{n} P_{r_i}^2 - \left(\sum_{i=1}^{n} P_{r_i}\right)^2} \tag{2}$$

$$a = \frac{\sum_{i=1}^{n} P_{p_i}}{n} - b\frac{\sum_{i=1}^{n} P_{r_i}}{n} \tag{3}$$

The position is represented in 2D information, i.e., x and y coordinates.

In the linear regression model, P_r is the dependent variable representing the real position labelled as (X_r, Y_r). At the same time, P_p is the independent variable representing the pixel position marked as (X_p, Y_p). Therefore, the pixel position regression of both variables is described as follows:

$$X_r = a_1 + b_1 X_p \text{ and } Y_r = a_2 + b_2 Y_p \tag{4}$$

2.6 Robot Kinematics

The real-world coordinate (Xr, Yr) of the detected object represented the arm robot's grasping point, that is, the target position of the arm robot's end-effector. The wrist determined the orientation of the end-effector. As described before, the arm robot was developed using 3 DOFs, i.e., base, shoulder, and elbow joints, which determined the 3D position (x, y, z) as shown in Figure 2. The input parameters of inverse kinematics are the real positions of the target can be expressed as x=Xr, y=Yr, and z is the height of the end-effector to the ground. When the objects were placed on the ground, their height positions were the same. Therefore, the height of z was initialized as z=0. Finally, the grasping point P was defined as P (X_r, Y_r, 0).

Inverse kinematics [14] was used to determine the angle of each joint. The target position (end-effector) must be known. The main 3 DOFs: base, shoulder, and elbow, were marked as θ_1, θ_2, and θ_3, respectively. The base was a joint used to move the arm robot body to turn left or right, the shoulder joint set the stretch movement of the arm, and the elbow worked on the up and down movement of the arm. Based on Figure 2, the end-effector P (X_r, Y_r, Z_r) was transformed into joints' angle, i.e., the θ_1, θ_2, and θ_3, which can be calculated using the equations as follows:

$$\theta_1 = \tan^{-1}\left(\frac{Y_r}{X_r}\right) \tag{5}$$

$$\theta_2 = \tan^{-1}\left(\frac{Z_r}{AD}\right) + \cos^{-1}\left(\frac{l_1^2 + AC^2 - l_2^2}{2 l_1 AC}\right) \tag{6}$$

$$\theta_3 = \cos^{-1}\left(\frac{l_1^2 + l_2^2 - AC^2}{2 l_1 l_2}\right) \tag{7}$$

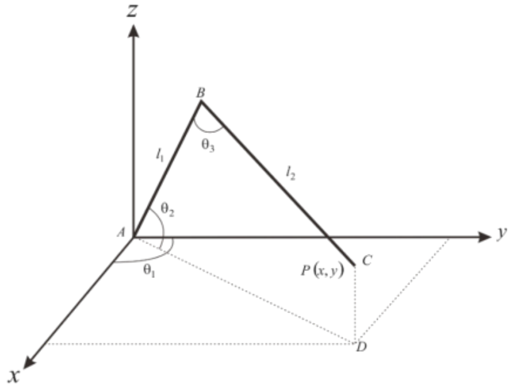


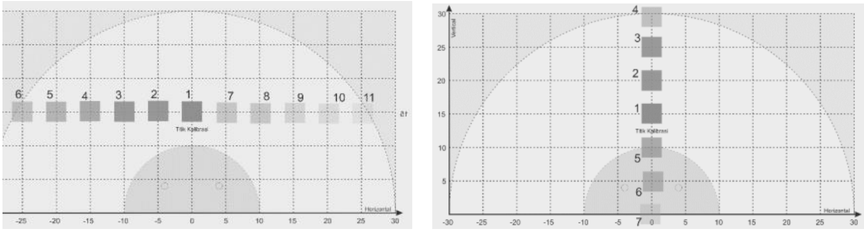
Figure 2: Three-DOF Arm Robot

3.0 RESULT AND DISCUSSION

The experiment observed the Pixy camera's performance in detecting the object. There were two experiment steps. The first step was to see the performance in recognizing object position using the pixel position regression method to estimate the objects' real-world position. The real position was identified and included in the kinematics equation to decide the angle of each DOF. The second step was to analyze the performance of the arm robot when grasping, picking up, and repositioning the object based on the image data of the Pixy camera.

3.1 Object Recognition Position

In this experiment, the method performance was observed to interpret the real-world positions based on the Pixy camera information. A single coloured object was placed in the detection area, divided into several sampling points. The sampling data was obtained from the represented x-axis and y-axis, divided every 5 cm from the centre point of the area (0,15) cm. The sampling point is presented in Figure 3. The closer the sampling data, the more accurate data to be obtained. The number of sampling points, defined as n, represents the data used in the linear regression. Here, the number of sampling points on the x-axis was 11, while the y-axis was 7. For each sampling point, the X_p and Y_p pixel data were taken 20 times, and the average values X_p in each point and Y_p in each point were generated to be used in the regression.



(a) (b)

Figure 3: (a) Sampling Point of *X* position, (b) Sampling Point of *Y* Position

The results of pixel position identification are shown in Table 1 and 2. Table 1 presents the result of pixel x (X_p), and Table 2 presents the result of pixel y (Y_p) detection. The pixel position of X_p and Y_p is the average of 20 data taken for each sampling point. The pixel position is used as the independent variable of the linear regression. The regression parameters of a_1 and b_1 were calculated using Equation 3. It generates the formula as shown follows:

$$X_r = 26.67 - 0.167X_p \text{ and } Y_r = -3.57 + 0.17Y_p \quad (8)$$

Table 1: Average of x-axis position in real-world (Xr)

No	Sampling Point of x	X_p	X_r	e
1	-25	300	-23.43	1.57
2	-20	281	-20.26	0.26
3	-15	254	-15.75	0.75
4	-10	225	-10.91	0.91
5	-5	192	-5.39	0.39
6	0	158	0.28	0.28
7	5	127	5.46	0.46
8	10	95	10.81	0.81
9	15	63	16.15	1.15
10	20	37	20.49	0.49
11	25	20	23.33	1.67
Average				0.79

The charts in Figure 4(a) and 4(b) show that the pixel regression has successfully interpreted the real object position based on the pixel position information as indicated by the average error value of X_r and Y_r, which are 0.79 and 0.35, respectively, and the highest error value is less than 2 cm. Based on Figure 4(c) and 4(d), the X_r and Y_r error values get higher when the object is far from the center point as it is affected by the lens of the Pixy camera, which is slightly convex. The results were compared with the research of [11], which discussed a similar topic and found that the average accuracy was between 1.42 cm to 5.93 cm. Hence, the average error in this study generated a better value because the error range was 0.04 cm – 1.67 cm. However, it should be considered that this experiment was conducted on a smaller area size (30 cm × 60 cm) compared to 140 cm × 200 cm in research [11].

Table 2. Average of y-axis position in real-world (Y_r)

No	Sampling Point of y	Yp	Yr	e
1	30	191	28.9	1.10
2	25	169	25.16	0.16
3	20	140	20.23	0.23
4	15	109	14.96	0.04
5	10	79	9.86	0.14
6	5	48	4.59	0.41
Average				0.35

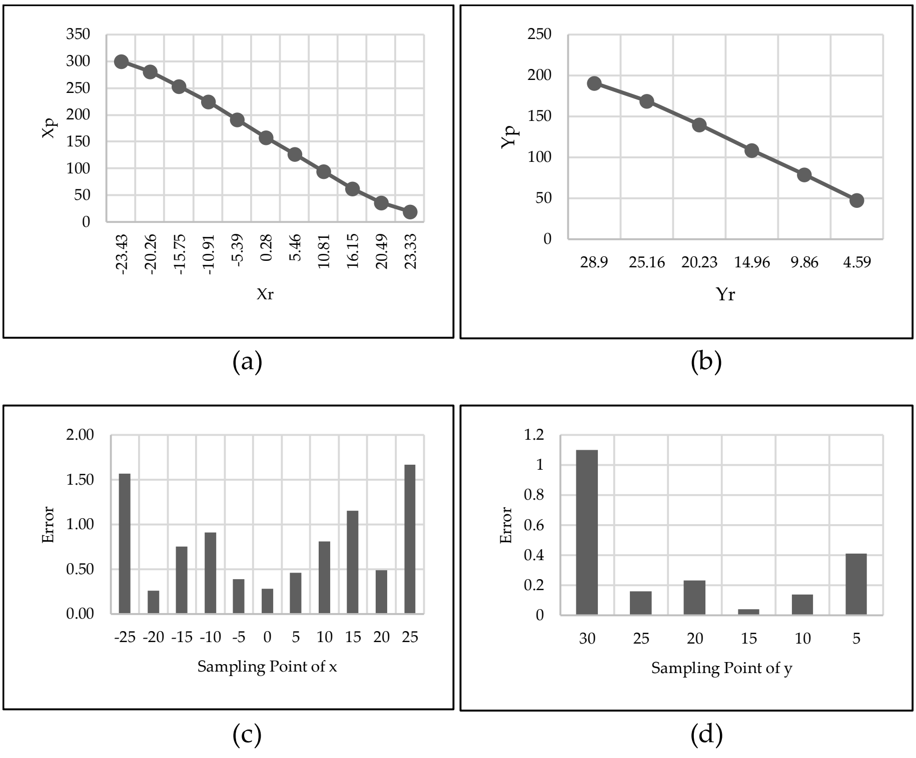


Figure 4. Experimental Results, (a) x-axis regression, (b) y-axis regression (c) Error rates of x, and (d) Error rates of y

3.2 Object Picking Using Arm Robot

Then, this research observed the performance of pixel position regression in the arm robot. The real object position (X_r, Y_r) was used as the input information of inverse kinematics, which arranged the movements of the arm robot by calculating the angle of each arm robot's joint. The θ_1, θ_2, and θ_3 were the angles generated from the position information $P(X_r, Y_r, Z_r)$. As described before, to perform the grasping and positioning task, the value of Zr was set to 0.

During the experiment, the object was placed randomly on the left side of the area. The Pixy camera detects the position by identifying the object's color ID and pixel positions. The real-world position X_r and Y_r were calculated using a linear regression formula. Then, the arm robot performed the task of grasping, picking, and repositioning the object into a specific area according to the object's ID.

Table 3: Angle Joint Calculation

No	x	y	z	θ_1	θ_2	θ_3	Results	Percentage
1	-21	2	0	-84.56	49.20	86.97	Success	100%
2	-22	7	0	-88.18	46.62	91.14	Success	100%
3	-23	6	0	-88.44	46.62	91.13	Failed	0%
4	-15	0	0	-90.00	64.94	58.13	Success	100%
5	-12	18	0	-33.69	47.69	89.15	Success	100%
Average								80%

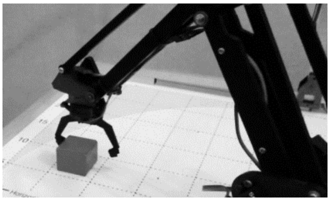


Figure 5: Example of robot motion to detect, grasp, and reposition an object

Table 3 shows the performance of the inverse kinematic in determining the angle of arm robot joints and the results of the object-picking task performed by the arm robot. Implementing the algorithm of pixel position regression showed that the picking task was done successfully in four out of five trials, which meant that the arm robot's success rate was 80%. Even though the Pixy camera has a fisheye effect, the algorithm implementation performed well compared to another method of a more complex algorithm, which needs a huge computational [18]. Figure 5 shows the arm robot's movements that successfully pick and move the object from the grasping point to the new position in real time. These results confirmed that the pixel position regression could be implemented in the arm robot to perform several tasks, i.e., grasping, picking, or repositioning the objects.

4.0 CONCLUSION

The pixel position regression method was applied to estimate the objects' real-world position based on the Pixy camera information. The method was tested on the arm robot to perform grasping, picking, and repositioning of the objects with a success rate of 80%. Furthermore, the error value from implementing this method ranged between 0.04 cm to 1.67 cm. Thus, it can be concluded that the pixel position regression method by using the Pixy camera was effective in being implemented in the arm robot to grasp, pick, and reposition the objects.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Higher Education, Indonesia, at Universitas Negeri Malang.

REFERENCES

[1] B.I. Kazem, A.I. Mahdi and A.T. Oudah, “Motion planning for a robot arm by using genetic algorithm”, *Jordan Journal of Mechanical and Industrial Engineering*, vol. 2, no. 3, pp. 131–136, 2008.

[2] 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, 2019.

[3] G.O. Tirian, “Automation of a warehouse by means of a robotic arm”, *Annals of Faculty Engineering Hundeoara–International Journal of Engineering*, vol. 11, pp. 267-271, 2013.

[4] M.A.A. Rahman, N.S. Osman, and C.H. Boon, “Configuring Safe Industrial Robot Workcell in Manufacturing Industry”, *Journal of Advanced Manufacturing Technology*, vol. 10, no. 2, pp. 125–136, 2016.

[5] K. Ghadge, S. More, P. Gaikwad and S. Chillal, “Robotic arm for pick and place application”, *International Journal of Mechanical Engineering and Technology*, vol. 9, no. 1, pp. 125–133, 2018.

[6] Z. Shangguan, L. Wang, J. Zhang and W. Dong, “Vision-Based Object Recognition and Precise Localization for Space Body Control”, *International Journal of Aerospace Engineering*, vol. 2019, pp. 1-10, 2019.

[7] G. Koutaki and K. Uchimura, “Fast and robust vision system for shogi robot”, *Journal of Robots and Mechatronics*, vol. 27, no. 2, pp. 182–190, 2015.

[8] T. Umeda, K. Sekiyama and T. Fukuda, “Vision-based object tracking by multi-robots”, *Journal of Robotics and Mechatronics*, vol. 24, no. 3, p. 531, 2012.

[9] R. Mitsudome, H. Date, A. Suzuki, T. Tsubouchi and A. Ohya, “Autonomous mobile robot searching for persons with specific clothing on urban walkway”, *Journal of Robotics and Mechatronics*, vol. 29, no. 4, pp. 649–659, 2017.

[10] O.Y. Ismael and J. Hedley, “Development of an omnidirectional mobile robot using embedded color vision system for ball following”, *American Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 22, no. 1, pp. 231–242, 2016.

[11] M.S.M. Aras, T.P. Chaing, M.K. Aripin, M.K.M. Zambri, A. Khamis, M.B. Bahar, and M.Z.A. Rashid, "Monitored and controlled underwater scissor arm manipulator using pixy camera", *Indian Journal of Geo-Marine Sciences*, vol. 48, no. 7, pp. 1120–1131, 2019.

[12] S. Lee, G. Tewolde, J. Lim and J. Kwon, “Vision based localization for multiple mobile robots using low-cost vision sensor”, *International Journal of Handheld Computing Research*, vol. 7, no. 1, pp. 12–25, 2016.

[13] J. Wang, S. Liu, B. Zhang and C. Yu, “Inverse kinematics-based motion planning for dual-arm robot with orientation constraints”, *International Journal of Advanced Robots Systems*, vol. 16, no. 2, pp. 1–14, 2019.

[14] M. Kalyoncu, “Mathematical modelling and dynamic response of a multi-straight-line path tracing flexible robot manipulator with rotating-prismatic joint”, *Applied Mathematical Modelling*, vol. 32, no. 6, pp. 1087–1098, 2008.

[15] R.R. Serrezuela, A.F.C. Chavarro, M.A.T. Cardozo, A.L. Toquica and L.F.O. Martinez, “Kinematic modelling of a robotic arm manipulator using Matlab”, *ARPN Journal of Engineering and Applied Sciences*, vol. 12, no. 7, pp. 2037–2045, 2017.

[16] Y. He and S. Chen, “Advances in sensing and processing methods for three-dimensional robot vision”, *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, pp. 1–19, 2018.

[17] K.H. Zou, K. Tuncali and S.G. Silverman, “Correlation and simple linear regression”, *Radiology*, vol. 227, no. 3, pp. 617–628, 2003.

[18] M. B. Campos, A. M. G. Tommaselli, L. F. Castanheiro, R. A. Oliveira, and E. Honkavaara, “A Fisheye image matching method boosted by recursive search space for close range photogrammetry”, *Remote Sensing*, vol. 11, no. 12, pp. 1-18, 2019.