

# Visual Servoing of Mobile Microrobot with Centralized Camera

Gandjar Kiswanto, Mohamad Safhire, Reza Afrianto and Rifkie Nurcahya

Department of Mechanical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia 16424

**Abstract.** In this paper, a mechanism of visual servoing for mobile microrobot with a centralized camera is developed. Especially for the development of swarm AI applications. In the fields of microrobots the size of robots is minimal and the amount of movement is also small. By replacing various sensors that is needed with a single centralized vision sensor we can eliminate a lot of components and the need for calibration on every robot. A study and design for a visual servoing mobile microrobot has been developed. This system can use multi object tracking and hough transform to identify the positions of the robots. And can control multiple robots at once with an accuracy of 5-6 pixel from the desired target.

## 1 Introduction

The role of robots in human life, especially in the manufacturing sector continues to increase and more relevant [1]. Today's robot system has reached the point where the system can run automatically and improve performance by interacting with the environment [2].

To do this, robots are equipped with various sensors to be able to understand the environment. But in the applications of microrobot systems that scale from cm to mm the various sensors are difficult to use because of their limited size and the required precision level of the sensor will make it difficult in the calibration process [3]. Visual Servoing or visual control is widely used for various applications of robot manipulator industry and mobile robot, starting from positioning, navigation, object tracking etc. [4] [5] [6]. Microrobot systems thus can benefit from this system as well.

In the case of mobile robots, it is rare to purely use visual control system to get information. Some methods use lasers to obtain distance information with objects around and various combination of sensors [7][8]. With the development of computer vision in general visual control system becomes more cheap, lightweight, compact and practical. Not to mention the amount of information that can be taken from one sensor. This makes the visual control system more attractive.

Visual control is very suitable for use in microrobot to replace the various sensors needed to know the position. This can make the robot system simpler because the inherent unbooted sensor allows us to minimize and simplify the robot design [9]. Visual control also allows us to control more than one robot with only one sensor [10]. With visual controls, we can control the robot large number or swarms of robots. Applications of such

systems include micro handling and assembly, micro sensing and one of the most unique of allowing us to simulate the behavior of swarm animals such as fish and bees [11] [12]. Of course, to be applied like this requires a reliable system visual control system. Therefore, it takes a visual control system that can adapt to environmental changes and can process data quickly so that the movement of the robot can be controlled almost real time.

## 2 System modeling

The system we want to create is where the robot feature will measure with a centralized camera in the form of Logitech HD C270 webcam. Data from this camera will be processed through the controller to be the position data that will be processed by the server to serve as a command to the robot. This command will then be provided via WIFI to each robot.

Control system in use is a closed loop system that compares the features obtained of what we call as ( $s$ ) with the desired feature ( $s'$ ). The purpose of this control system is to minimize errors. And **Figure 1** is there to explain this in a better form.

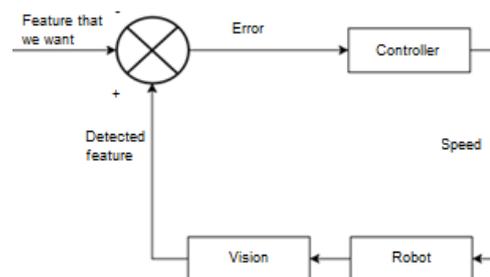


Figure 1. Control Diagram for Microrobot.

Featured in this case is a 2D single point information taken from a color camera picture. Thus, error can be defined as:

$$e(t) = s(t) - s' \quad (1)$$

$s(t)$  defines features detected during ever-changing movements.  $S'$  is an input that does not change with time. Here  $s(t)$  is the position and orientation data in cartesian coordinates in the image.

$$s(t) = \begin{bmatrix} m(t) \\ n(t) \\ \theta(t) \end{bmatrix} \quad (2)$$

The kinematics model of robots moving in 2D with 2-wheel system can be lowered to.

$$\dot{m} = v \cos(\theta) \quad (3)$$

$$\dot{n} = v \sin(\theta) \quad (4)$$

$$\dot{\theta} = \omega \quad (5)$$

Or can be written as

$$\begin{bmatrix} \dot{m} \\ \dot{n} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (6)$$

With that in mind if we want to bring the robot towards a position  $(m', n')$  with an orientation of  $\omega$  we can find the speed that is needed as

$$v = K_v \sqrt{(m - m')^2 + (n - n')^2} \quad (7)$$

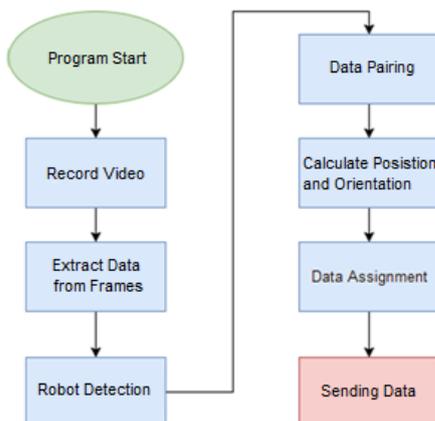
$$\omega = \tan^{-1} \frac{n' - n}{m' - m} \quad (8)$$

The above equation then describes the movement of the robot against the speed that is given. This equation will then be used to obtain the amount of error that we need to compensate with the vision system.

### 3 Vision System

This section explains how the vision system detect the robot, assigns the label to the robot, and calculates the position and orientation of the robot. An overview of the entire vision system can be seen in **Figure 2**.

#### 3.1 Robot detection



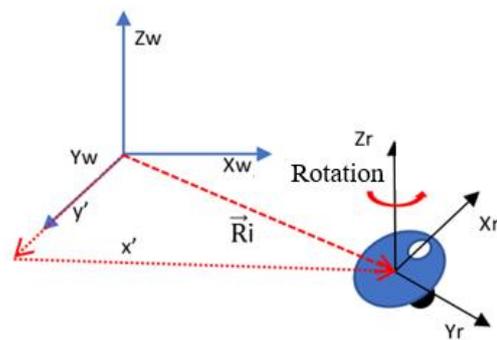
**Figure 2.** Flowchart for the Vision System that is used.

Robot detection is done using Circular Hough Transform (CHT). CHT is a derivative of the basic form of Hough Transform. This approach is used because of its toughness with noise, occlusion and varying lighting. The foreground pixel with a high gradient is designated as a candidate pixel and allowed to give 'noise' to the accumulator circuit. In the application of classical CHT, candidate pixels vote in patterns around them that form a full circle of fixed radius [11]. The data obtained in the form of coordinate position data to the camera axis. This data is a 2x1 matrix. But the data obtained has not been paired to find the angle. This will be explained in the next section.

#### 3.2 Data pairing, position & orientation calculation

Data pairing is used to pair the data between the center point of the robot circle and the reference center point of the circle data. This needs to be done to calculate the orientation angle of the robot. Data pairing is done by finding the smallest distance between the reference circle to the central point of the robot circle. The data from CHT discussed earlier will be divided into 2 sections. The first part is to circle the robot in the form of a 2xn matrix from the x and y coordinate positions of each circle. The second part is similar but for the coordinate positions of the x and y circles of reference. Data from these two matrices will be paired with the point closest to it. The data in the robot circle matrix is used as a reference.

After the data has been paired we can calculate the orientation of the robot. Orientation data is the result of angle calculation between the robot center point and the reference point. By knowing the distance between the origin point and the center of the robot and the reference point we can know the orientation of the robot as **Figure 3** below illustrates.



**Figure 3.** Position and Orientation of the robot.

If we give position notation of robot circle as X and Y and position of reference circle as M and N then orientation can be formulated to be:

$$\phi = \tan^{-1} \left( \frac{(N-Y)}{(M-X)} \right) \quad (9)$$

Then this process is done for all robots to get the orientation data and position. The final data that is in the form of a 5xn matrix in which n states many robots are

detected at that time. Though this data still needs to be labeled to know which robot has this data. For that the part below will explain.

### 3.3 Data Assignment

The data obtained above is the data that we still cannot identify. Because the data above is not yet indexed and cannot describe which robot has that data. What we need is that the data of robot no 1 to not be confused with the data for robot no 2 and so on. To do so we compare the data that we have now ( $T$ ) with the data in the previous frames ( $T-1$ ). And by using a queue system where the first robot that enters will be labeled robot no 1, the next robot no 2 and so on. We can be sure that each robot data will not be confused with another robot.

New problems will arise when there is a robot that disappeared suddenly or occluded by an object. To solve this problem, we gave a special algorithm when there is an addition or reduction of a robot. When the number of robots detected now ( $T$ ) is less than the previous robot data ( $T-1$ ) the robot location data that has is missing lost will not be updated until the robot appears again. This algorithm adds to the robustness of the system when there is an error. The output of this can be seen in **Figure 4**.

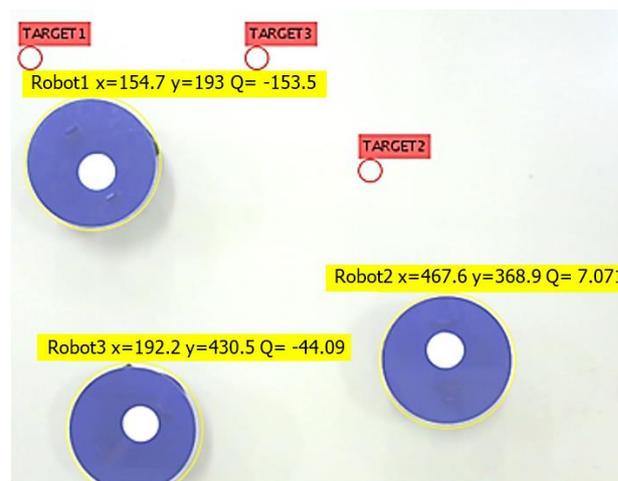
### 4 Control Algorithm

The control algorithm of the visual servoing will only match the features we want to the ones detected. To achieve this, we need to breakdown into a few simple commands. Due to the robotic construction and also the limitation of the type of movement that can be done we divide it into 4 simple commands that is stop, forward, turn clockwise and counterclockwise. From these 4 commands, we are able to run the robot.

The initial stage is to get the position data from the server the target position that we want. From these two data, we can calculate the amount of angle we need. The robot will then rotate between to the right or to the left to equalize the angle of orientation that it requires. When the robot has equalized its angle, it will be given the orders to move forward. As the robot moves to the position the robot will continue to be updates its location to know the remaining distance that must be taken. When the remaining distance has reached the limit threshold that we have given or what we say as the robot's tolerance it will stop and at this stage the robot position will be the same as the feature we want.

To perform complex movements such as spline circle etc. we can solve the movement to 20 linear motions by way of interpolation. Given the robot can only move linearly then to reduce the error need to use interpolation. Interpolation takes the 20 coordinate points that are in the path we want and include them as target data one by one. With that we can command the robot to move in a complex way without large errors.

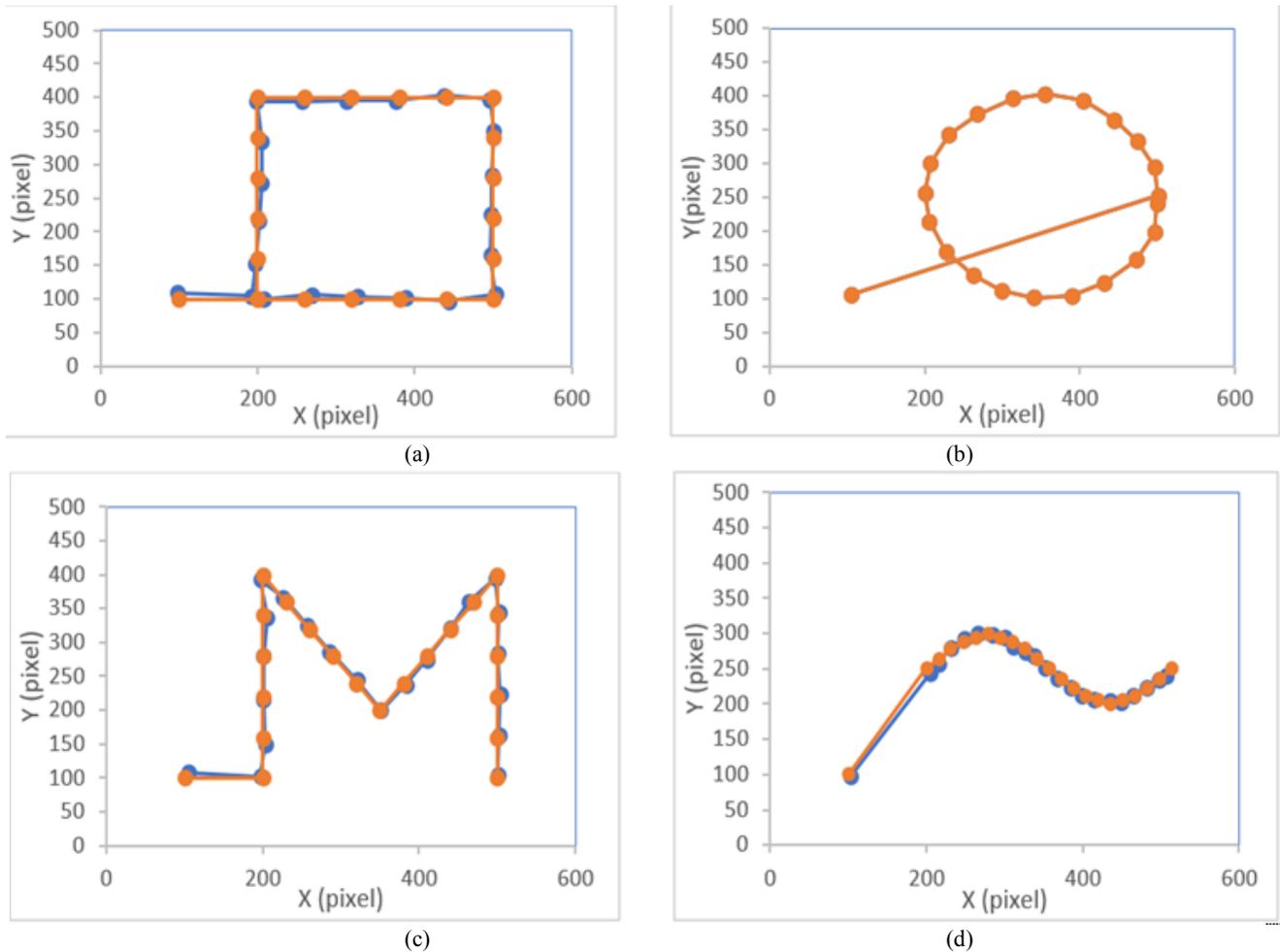
The various kinds of motion that we have made is the movement of circle, sinus, waypoint, and point to point. From these various movements, we will test how precise the robots can be moved. By the method described in the next section.



**Figure 4.** Display of Control System.

### 5 Experimental testing

We divided our experiments into 4 types of movements in which each will test the motion capabilities that the robot can perform. The four types of movements that are shown in **Figure 5** are squares, "M", circles and spline. Our square shape in **Figure 5** (a) is used to test linear motion of robot's ability. Circle shape in **Figure 5** (b) tests the ability of circular motion. The form "M" in **Figure 5** (c) tests the ability to move in 2 axes at the same time. And spline movement in **Figure 5** (d) to test the fluidity of the movement.



**Figure 5.** Experimental result (a)Square (b)Circle (c)M (d)Spline.

The result of the robot movement that we analyze is the location of the robot compared to the 20 targets that we command. Retrieval of data on the vision system is done every moment. But because the robot model cannot change direction instantly the data that we will be analyzing is in the moment the robots has achieved its target before it changes its direction to go to the next target. This data is illustrated in **Figure 5**.

As described before we have divided the experiment into 4 groups that is (a)Square (b)Circle (c)M (d)Spline. A total of 22 robot position for each shape are tabulated and graphed. This is done to compare between the data in the form of targets and actual data of the position from the robot. From both data, we calculate the resultant and deviation from the target coordinate in mm. The value gained from the experiment in mm will be shown in **Table 1**. A slight deviation in the starting position like we can see in **Figure 5** is to be expected therefore it is not calculated in the results.

**Table 1.** Experimental Results in mm

Shape	Maximum	Minimum	Average
Square	9.440	3.092	6.082
Circle	8.975	1.935	5.863
M	8.900	0.795	4.511
Spline	9.791	1.422	4.344

An interesting phenomenon that we found is with the simple shapes like in **Figure 5** (a) there is quite a lot more deviation compared with the more complex shapes. We concluded that this is due to the distance between the targets. With higher distances, it becomes harder to self-correct. Statistical analysis is done to get the precision value from our robot.

From the 4 experiments as many as 84 data in then joined to get the average data value of 5.2 mm and standard deviation of 2 mm. The accuracy of the robot movement in this case can be calculated by knowing the magnitude of the deviation of the movement of the robot. For that we need to know the average size of the robot deviation population. Using the average interval, we can know that the range of the average population is between 4.77-5.63 mm. With that we can say that our robot can move to the destination that we command with an accuracy of 5.63 mm.

## 6 Conclusion

A Visual control system for mobile microrobot with a centralized camera was created. This vision-based control system is able to control multiple robots at once by comparing the features detected with the desired ones. By using a webcam sensor, we can get the location data and orientation of each robot. Labeling on each robot can be done and there are no errors in labeling such as the miss interpretation of data or others. The vision system is also resistant to external disturbances such as occlusion and light changes. This can be done because it has a robust algorithm in case of interference from the outside. The robot can be command using a control system where it can perform simple movements to create geometric shapes. The control system can move the robot to its destination with an accuracy of 4.77-5.63 mm from the target. All systems such as the robot, server, and control systems have been well integrated to be able to drive the robot. Further study is also needed to determine the optimal distance between targets to achieve better accuracy. Therefore, in a later study the variable of target distance should also be considered.

## References

1. K. M. N. T. J. Ogbemhe, "Achieving Sustainability in Manufacturing Using Robotic Methodologies," *Procedia Manufacturing*, p. 440–446, (2017)
2. M. A. A. K. J. B. K. Hamed Shahbaz, "Demonstration learning of robotic skills using repeated suggestions learning algorithm," *Biologically Inspired Cognitive Architectures*, p. 21–30, (2017)
3. C. A. ., N. H. A. C. R. N. Michael Rubenstein, "Kilobot: A low cost robot with scalable operations designed for collective behaviors," *Robotics and Autonomous Systems*, p. 966–975, (2014)
4. G. C. F. S. F. C. Don Joven Agravante, "Visual Servoing in an Optimization Framework for the Whole-Body Control of Humanoid Robots," *IEEE Robotics and Automation Letters*, pp. 608-615, (2017)
5. S. K. P. P. Gossaye Mekonnena, "Wireless hybrid visual servoing of omnidirectional wheeled mobile robots," *Robotics and Autonomous Systems*, pp. 450-462, (2016)
6. M.-Y. C. Wei-Che Chang, "Image feature command generation of contour following tasks for SCARA robots employing Image-Based Visual Servoing—A PH-spline," *Robotics and Computer-Integrated Manufacturing*, pp. 57-66, (2017)
7. D. H.-P. H. Martinez-Barbera, "Autonomous navigation of an automated guided vehicle in industrial environments," *Robot. Comput.-Integr. Manufacture.*, pp. 296-311, (2010)
8. R. O. C. S. C. F. D. Ronzoni, "AGV global localization using indistinguishable artificial landmarks," dalam *Proceedings of IEEE International Conference on Robotics and Automation*,, Shanghai, (2011)
9. C. D. S. F. Daniel Jasper, "Hardware-based, Trajectory-Controlled," *IFAC Symposium on Mechatronic Systems*, (2010)
10. C. Diederichs, "Fast Visual Servoing of Multiple Microrobots using an FPGA-Based Smart Camera System," dalam *Proceedings of the 18th World Congress The International Federation of Automatic Control*, Milano, (2011)
11. Q. L. Y. & X. N. Xu, "Design, Fabrication, and Visual Servo Control of an XY Parallel Micromanipulator With Piezo-Actuation," *IEEE Transactions on Automation Science and Engineering*, pp. 710-719, (2009)
12. N. T. H. L. P. & Q. T. T. Thinh, "Research and Applying Computer Vision for Controlling the School of Fish Robots Using Swarm Model.," dalam *3rd International Conference on Green Technology and Sustainable Development*, (2016)
13. E. Davies, *Machine Vision: Theory, Algorithms, Practicalities.*, Morgan Kauffman Publishers, (2005)