Simulation and Application of Mobile Robot Vision Technology

Heru Suwoyo
Department of Electrical Engineering, Universitas Mercu Buana, Indonesia

Taufik Hidayat
Department of Electrical Engineering, Universitas Indonesia, Indonesia

Abdurohman
School of Mechatronic Engineering and Automation, Shanghai University, China

Minghao Yu
School of Mechanical and Electrical Engineering and Automation, Shanghai University, China

Abstract: This paper will focus on the application of machine vision in mobile robots and take moving to the appropriate position and grasping the designated target as the task. This paper will describe and simulate the vision technology that mobile robots need to apply in the task process. This paper mainly uses the camera as the sensor. The difficulties of vision technology are mainly divided into three parts: scene depth information acquisition, positioning and mapping, and image processing. In order to obtain the depth information of the scene, this paper mainly introduces the depth information acquisition methods of monocular camera and binocular camera. In the aspect of localization and mapping, this paper mainly introduces the simulation of visual odometer to understand the basic process of mobile robot obtaining navigation map and its own route. Then, the gray gradient 2D maximum entropy algorithm is introduced to segment the scene and target, and extract features to judge the required target. Compared with other segmentation algorithms, the gray gradient 2D maximum entropy algorithm has higher segmentation accuracy, but the operation time is longer. This paper has simply optimized its operation efficiency. Finally, this paper describes the positioning method of grasping the object with a two degree of freedom manipulator using the knowledge of inverse kinematics. Because of the epidemic situation, schools cannot obtain experimental equipment. This paper mainly demonstrates the effectiveness of the algorithm through simulation.

Keywords: Scene Depth Information Acquisition, Visual Odometer, Two-Dimensional Gray Gradient Maximum Entropy Algorithm, Inverse Kinematics Solution.

Correspondents Author:
Taufik Hidayat, Department of Electrical Engineering, Universitas Indonesia, Indonesia
Email : taufik.hidayat23@ui.ac.id

Introduction

The research object of this course is mobile robot, and my research direction is machine vision. This paper will mainly simulate and experiment the application of machine vision algorithm of mobile robot. Mobile robots have developed maturely since 1953, and many kinds have been derived. According to the classification of mobile equipment, mobile robots are mainly divided into wheeled robots, leg robots, crawler robots, etc. There are many kinds of mobile robots, but in the final analysis, the task they perform is to arrive at the designated place to detect or take items (Houtman et al., 2021). Therefore, this paper will mainly take a small transport robot as an example to simplify the task into arriving at a designated place to take an item and returning (Li et al., 2022). To complete this seemingly simple task, the robot first needs to be able to sense the surrounding environment, locate its relative position in the environment, then control its movement according to the target position through path planning, detect the position of the target object to be picked up, obtain the target and return (Lin et al., 2021). As shown in Figure 1, the mobile robot needs such a complex task to complete simple pickup and retrieve tasks. Machine vision is the basis for mobile robots to achieve the above steps. This paper will mainly simulate and explain the application of the above vision technology (Nwachioma & Pérez-Cruz, 2021).

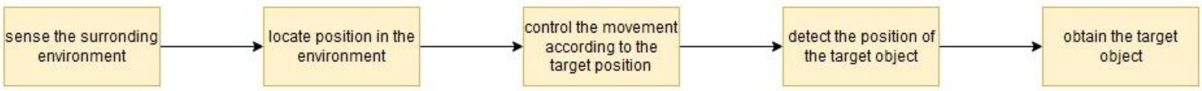

Figure 1 Mobile Robot Task Flow Chart

Research Method

Research Prospect and Purpose

Mobile robot is a kind of industrial robot, involving sensor technology, information processing technology, automatic control technology, navigation technology, etc. It can be used for logistics transportation under complex working conditions of intelligent manufacturing and autonomous navigation in outdoor environment and has broad application prospects (Tengilimoglu et al., 2023). At present, the most important applications of mobile robots are household robots and transportation in unmanned factories (Suwoyo et al., 2022). The foundation of these two parts of work is to move and take the designated item mentioned in Figure 1. Therefore, when the mobile robot can complete the simulation of the task of moving and taking the item, it proves that it can meet most of the work requirements. This paper aims to try a series of simple and feasible algorithm research and simulation applied to the vision of the task of the mobile robot (Suwoyo & Harris Kristanto, 2022).

Research Contents

Based on the understanding of the task content and significance of mobile robots, the following main research contents are as follows:

(1) The second chapter will mainly describe the hardware architecture of the mobile robot in this paper, such as mobile devices, manipulators, sensors, etc., and will mainly introduce the methods of different types of camera sensors to obtain the distance information in the image.

(2) The third chapter mainly descrexplain application of the visual odometer in slam and carries out simulation to obtain a simple camera track, and explains its principle, such as feature points, least square method, etc., as well as necessary prior operations such as camera calibration.

(3) The fourth chapter mainly introduces the two-dimensional gradient maximum entropy algorithm to improve the accuracy of target object recognition. The idea of the algorithm is to establish the symbiosis matrix according to the gray level and gradient of each point of the image as the horizontal and vertical coordinates.

(5) The fifth chapter will introduce how the two degree of freedom manipulator moves its own manipulator to grasp the target object through the inverse kinematics solution and introduce the ideal and actual working range of the two degree of freedom manipulator.

(6) The sixth chapter will summarize the work of this paper, analyze the innovation and shortcomings of this work, and discuss the direction of further research.

Hardware System

In order to enable the mobile robot to complete the tasks of navigation and move, identification and grasping, the hardware system of the mobile robot needs to be composed of three parts: mobile system, manipulator system and sensor system (Choudhary et al., 2022).

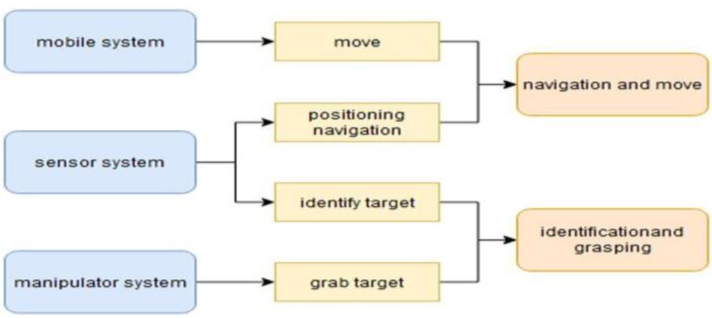

Figure 2 Functional Diagram of Mobile Robot System

Mobile System

The mobile system of mobile robot is mainly divided into three types: crawler, wheel and leg, which have their own advantages and disadvantages and applicable environment. The advantages of wheeled robot are fast speed, high efficiency and low motion noise, but its disadvantages are low turning efficiency, poor terrain adaptability and relatively flat environment. Crawler robot has the advantages of obstacle surmounting ability, strong terrain adaptability, and can turn on the spot, but its disadvantages are slow speed, low efficiency, high motion noise, and it is suitable for the environment with many obstacles in the field. Advantages of Legged Robots (Luo et al., 2023). Because this paper only needs to carry out a simple task simulation of taking target object, and the main environment is flat ground, as shown in figure 3, this paper will use four wheels as the mobile system, two front wheels as the driving wheels, respectively connect the motor, judge the distance obtained through the sensor system and use PID to control to carry out the mobile task (Shafaei & Mousazadeh, 2023). When it is necessary to turn, the two motors carry out differential control, although it will bring the disadvantage of large turning radius, But the control of wheeled robot is more convenient, and the speed is higher.

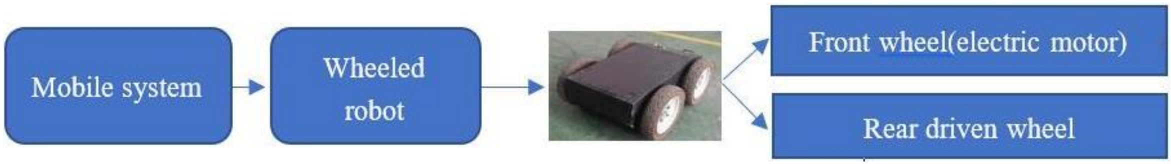

Figure 3 Schematic Diagram of Mobile System

Manipulator System

A manipulator system is an automatic operating device that can imitate human hands and arms to grasp, transport objects or operate tools according to fixed procedures, and is also a necessary system for carrying out the task of taking. After the sensor system detects and obtains the target coordinates, the manipulator system can obtain the movement and rotation required by each mechanical joint when taking the object and perform the grasping task through the inverse kinematics solution. Since there are six degrees of freedom in the space, the upper limit of the general manipulator's degrees of freedom is six. Because the mobile system is adopted in this paper, when the mobile robot moves to the same plane as the target, it can easily grasp the target object with only two degrees of freedom and reduce the complexity of the inverse kinematics solution. The manipulator with two degrees of freedom is shown in Figure 4, where the O point and the intersection point of the manipulator l1 and l2 are respectively equipped with motors, so that the included angle between l1 and x axis, l2 and l1 can be achieved to grasp the target on the plane (Teck & Dewil, 2022).

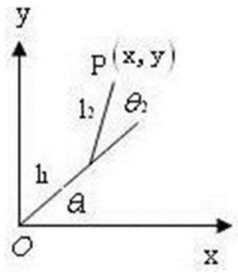

Figure 4 Mathematical Model of Two Degree of Freedom Manipulator

Sensing System

For mobile robots, the two biggest problems that visual navigation needs to solve are: Where am I which is a self-positioning problem. What is the surrounding environment which is Environmental modelling (Geng et al., 2022). The basic hardware to solve these two problems is the sensor, the sensors are mainly divided into laser sensors and cameras. Their advantages and disadvantages are summarized in following table.

Table 1 Advantages and Disadvantages of Radar and Camera

Lasers	**Cameras**
accurate	cheap
fast	rich information
heavy	high computation cost
expensive	work under assumptions

As radar equipment is expensive, we select the camera as the sensor.

Camera Characteristics and Types

The photos taken by the camera record the three-dimensional environment in the form of two-dimensional projection. The camera features that it can contain more information, but the camera lacks the depth dimension information. Therefore, the first problem for subsequent navigation and object distance recognition is how to obtain the depth information of the camera, which will be introduced in the next section. Cameras are mainly divided into monocular cameras and binocular cameras. The monocular camera only contains one lens, and the depth dimension information needs to be obtained by moving the camera. The binocular camera contains two lenses, and the scene depth information can be obtained through the sight distance (Xue et al., 2021).

Obtain Depth Information

The binocular camera consists of left and right cameras. We usually use parallax to estimate the depth of the target. Parallax refers to the difference between the positions of the same feature point in two images. Figure 5 is an aerial view from top to bottom, b is the distance between the left and right cameras of the binocular camera, xl corresponds to the points in the left camera image, xr corresponds to the points in the right camera image, and the difference between xl and xr is the parallax. Through the proportional relationship between similar triangles, we can get the depth relationship between the binocular camera and the target through the focal lengths f, b and parallax, as shown:

$$z = \frac{bf}{parallax} = \frac{bf}{xl - xr} \tag{1}$$

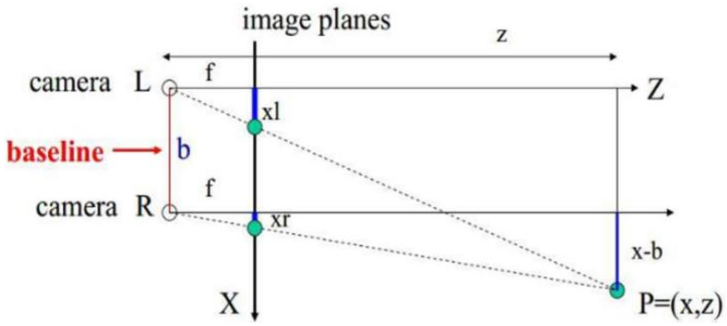

Figure 5 Binocular Camera Model

For a monocular camera, as shown in the Figure 6, we move the camera, where o_1 and o_2 are the light points before and after the monocular camera moves, and x_{1j} and x_{2j} are the same feature points on the photo, so when ignoring camera errors and distortion, the intersection point obtained by extending the straight line of o_1x_{1j} and o_2x_{2j} is the real target feature point x_j, so that we become a mapping from 2d photos to the 3d real world, and of course, we also obtain the depth information of the object.

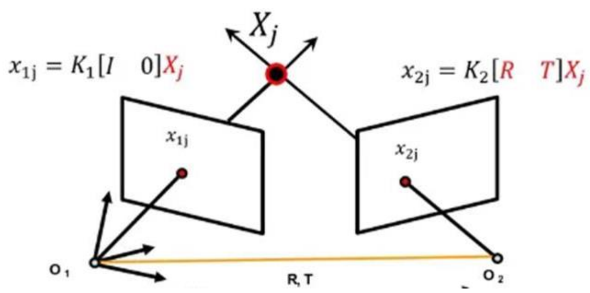

Figure 6 Method of Obtaining Depth with Monocular Camera

The above method is an ideal method for monocular camera to obtain depth information. However, due to camera distortion and error, the same intersection points found by the above method in multiple movements may not converge together. At this time, it is necessary to minimize the error by least square method and other methods to obtain the most accurate actual point to obtain accurate depth information. The sensor used in the subsequent simulation of visual odometer in this paper is a monocular camera (Ben Azouz et al., 2015).

Overall System Response Process

Based on previous discussed section, we can basically build the main platform structure of the mobile robot, and its response mechanism is shown in Figure 7.

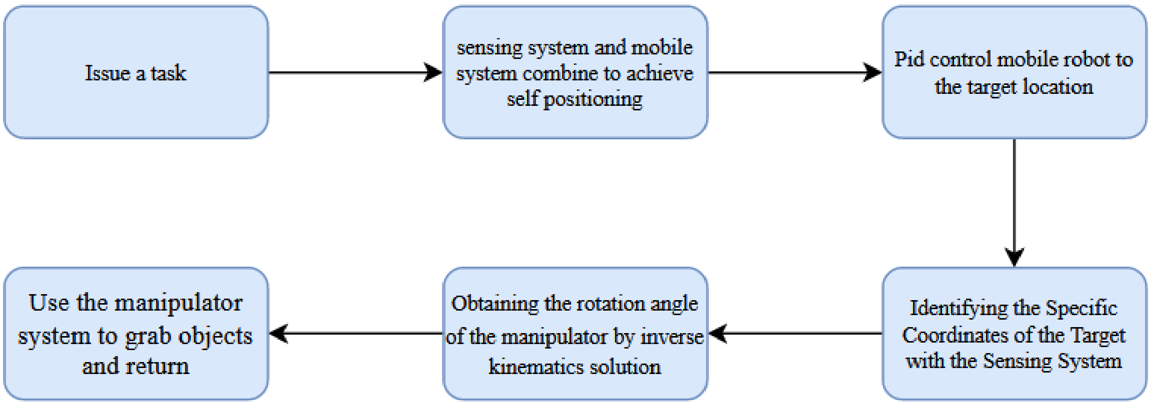

Figure 7 Overall System Response Process

Figure 7 shows the completion of such a simple capture task also requires the combined control of the three systems. However, due to the epidemic situation, we cannot go to the laboratory and other places, so this paper can only use the above as the hardware assumption for simulation, mainly introducing the algorithm control and analyzing the results (Arrais et al., 2017).

Result and Discussion

Monocular Visual Odometer

SLAM (simultaneous localization and mapping) is the full name of real-time positioning and map building or concurrent mapping and positioning. Its main function is to enable robots to complete localization, mapping, and navigation in unknown environments. A crucial problem in slam is how to determine the position and pose of the mobile robot itself, this problem is solved by visual odometer. The visual odometer takes different photos obtained during moving as input, and outputs the moving track of the camera through feature point matching. Since the camera itself will not rotate when the mobile robot moves in front of the target, the output camera track is also equivalent to the mobile robot's moving track.

Camera Calibration

In the photos taken by the camera, we can obtain the pixel coordinates of each object. However, in order to subsequently obtain the trajectory of the mobile robot in the real world, we need to master the corresponding relationship between the pixel coordinates and the actual coordinate system. Between the world coordinate system and the pixel coordinate system, there is also a transitional transformation relationship between the camera coordinate system and the image coordinate system. The specific relationship between them is shown in Figure 8.

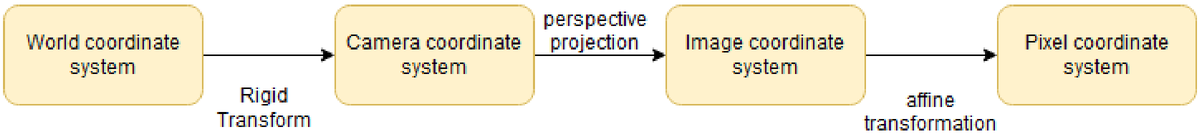

Figure 8 Coordinate System Diagram

In an ideal case, the relationship between the world coordinate system and the pixel coordinate system is as follows.

$$z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{dX} & \frac{cot\theta}{dX} & u_0 \\ 0 & \frac{1}{dYsin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} \tag{2}$$

In the above formula, (u, v, w) is the physical coordinate of a point in the world coordinate system, (u, v) is the pixel coordinate of the point in the pixel coordinate system, and z is the scale factor. In fact, the camera lens will inevitably produce radial distortion and tangential distortion, we need use the camera calibration to correct the lens distortions and generate a corrected image. The purpose of camera calibration is to find out the matrices in the above formula and the camera distortion parameters. I use a calibrator-based approach, which requires the shape and size of the calibration plate to be set. Therefore, we use a calibration plate with a grid spacing of 21.5 mm as the calibrator which is shown in figure 9.

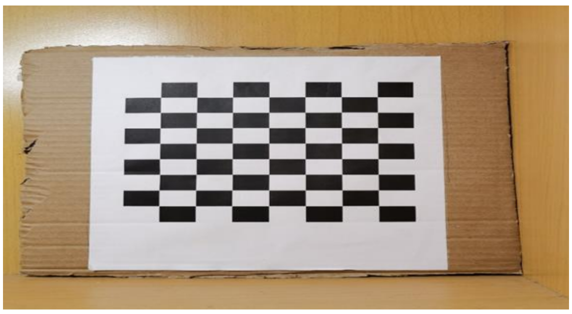

Figure 9 Calibration Board with Grid Spacing of 21.5 mm

Fortunately, matlab has the application of calibration, and we just need to upload the images we took, as shown in the figure 10 (the calibration board taken from different positions and angles; Add up to 20 photos) Matlab can load these pictures and compute the parameters we need through the methods of image processing and mathematical transformation.

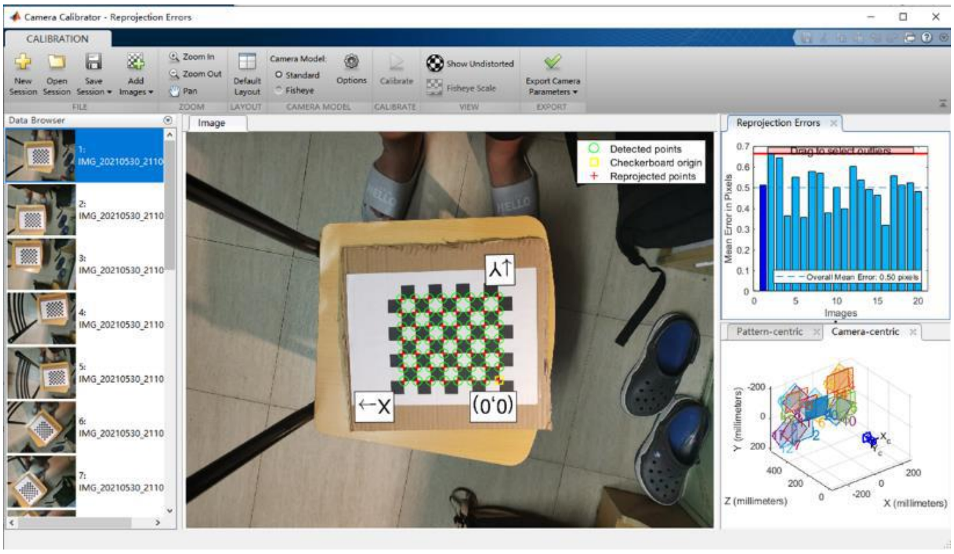

Figure 10 Application of Matlab Calibration Toolbox

Visual Odometer Simulation

To complete the monocular visual odometry, firstly, we should input the image sequences of cameras. Secondly, we take the step of feature detecting and matching. Finally, we use the local optimization to improve the estimation of motion. The specific flow chart is shown in the Figure 11.

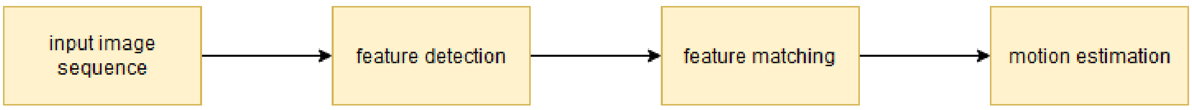

Figure 11 Flow Chart of Visual Odometer

Input Image Sequence

I chose Pikachu as the central object, so that the mobile robot could take pictures while surrounding the target. In order to improve the accuracy of shooting, I need to input as many pictures as possible, so this paper uses video mode to shoot the target, and finally decompose it into 150 photos. I use these 150 consecutive pictures as the input of image sequence.

Figure 12 Picture Input

Feature detection and matching

The meaning of the algorithm on the left is to estimate the pose of the second view by estimating the essential matrix and decomposing it into camera location and orientation.

(1) Convert picture to grey scale because after graying, the dimension of the matrix decreases, the operation speed is greatly improved, and the gradient information is still retained.

(2) Then match feature points and calculate the feature distance of adjacent images to get the similarity to estimate the pose.

(3) Store the correct feature points and repeat the above process.

Motion Estimation

In 2.3.2, when introducing the monocular camera to obtain depth information, this paper has introduced the basic idea of triangulation. Similarly, when we know the target coordinates, we can also obtain the camera moving path through triangulation. However, due to the existence of errors, in order to improve the calculation accuracy, this paper introduces the least square method. Because of errors, there are many pictures or cameras, and it is rare to find the actual point in Figure 6. Therefore, the least square method is needed to find the actual point. The main idea is to find a point in space. If the distance between it and each ray i is di, when the sum of di is the smallest, we think that the point is the actual point with the smallest error.

$$f(x, y, z) = min\left(\sum_{i=1}^{k} d_i\right) \quad (2)$$

Local Optimization

Due to the influence of local similarity, feature point matching often appears false positive, that is, feature points are matched with different similar feature points. In order to reduce this phenomenon like the Figure 13, I used the RANSAC algorithm to estimate the object plane to ensure that a more correct feature point pair is selected. The basic idea of the algorithm is as follows:

Plane equation refers to the equation corresponding to all points in the same plane in space. Its general form is $A\,x + B\,y + C\,z + D = 0$. The process of RANSAC plane fitting is as follows [3] :

(1) Randomly select three points in the initial point cloud and calculate their corresponding plane equations

(2) Calculate the algebraic distance from all points to the plane to select the threshold value. If di ≤ the threshold value, the point is considered as a sample point in the model, otherwise it is a sample point outside the model, and the number of current internal points is recorded.

(3) Repeat the above steps to select the best fitting parameters, that is, the model parameters corresponding to the plane with the largest number of interior points.

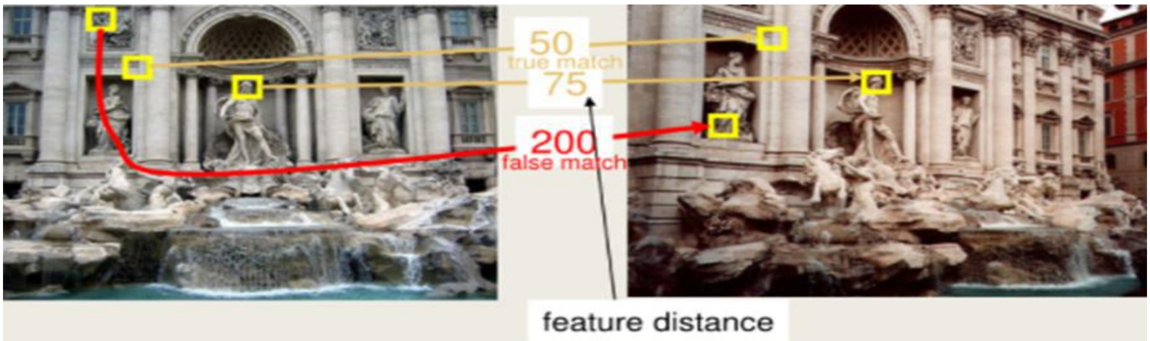

Figure 13 Feature Point Error Matching

Result of the Visual Odometer Simulation

The result of the visual odometer is shown in Figure 14. The closed loop trajectory in the figure also shows the accuracy of the experiment. However, due to the experimental error, the trajectory effect is not completely as expected, but this simulation also proves that the most basic mobile robot motion position estimation can be completed only with these basic algorithms.

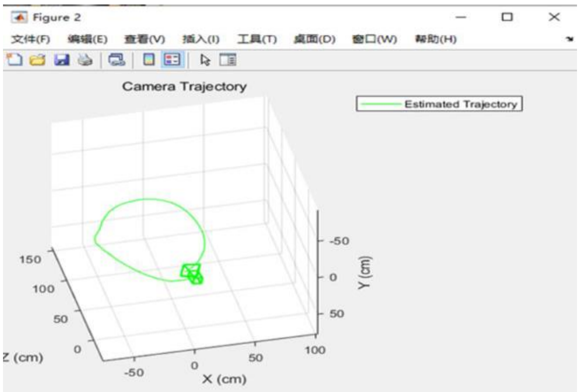

Figure 14 Moving Track of Mobile Robot

Image Segmentation

Image Segmentation Methods

When the mobile robot needs to grasp and locate the target, it usually uses image segmentation and feature extraction technology. This chapter will mainly describe the algorithm research and simulation of image segmentation. Image segmentation refers to the process of dividing an image into several regions that have certain regional consistency and do not overlap with each other. Mobile robots can extract the target region in the scene through image segmentation, and then extract the feature to find the final target location. Image segmentation methods mainly include segmentation methods based on threshold, edge, region and clustering. The following will introduce and compare the simulation results.

Otsu Method

Otsu method belongs to image segmentation algorithm based on threshold, and its core is to find appropriate segmentation threshold T. When Otsu method is used to divide all pixels of medical image into foreground (grayscale greater than T) and background (grayscale less than T) Assume that the average gray levels of the foreground and background are h1 and h2, and the probabilities of pixels in the foreground and background are p_1 and p_2, respectively. At the same time, the average gray level of the whole image is h. The principle of the method is as follows:

$$\begin{gathered} p_1 \times h_1 + p_2 \times h_2 = h \\ p_1 + p_2 = 1 \end{gathered} \tag{4}$$

The intra-class variance between foreground and background is:

$$\sigma^2 = p_1(h_1 - h)^2 + p_2(h_2 - h)^2 \tag{5}$$

Variance is the standard to measure data dispersion. Therefore, when the inter class variance between foreground and background is larger, it also proves that the more discrete the foreground and background are, the higher the irrelevance is, which also reflects the accuracy of segmentation from the side. The main idea of Otsu method is to find T to maximize the inter class variance between background and foreground. The step of Otsu method is to find the maximum inter class variance by traversing T, and then segment the image with T as the threshold.

Maximum Entropy Method for Image Segmentation

The maximum entropy method is also an image segmentation algorithm based on threshold, and the core idea of the maximum entropy method is to maximize the image entropy. Image entropy is a data form of image features, which mainly reflects the amount of information in the image. When the image entropy value is larger, it contains more information. Let the probability of the point whose gray value is i in the image be pi. In medical image segmentation, the gray entropy of graphics is mainly used to calculate, and the objective function is：

$$H = -\sum_{i=0}^{255} p_i log p_i \tag{6}$$

Region Growing Method

Region growing method is a semi-automatic region-based image segmentation method. For medical images, its main idea is to preselect seed points and grow outward to judge whether the gray value of adjacent points is less than the judgment threshold and gather the adjacent pixel points with close gray value. If the gray value of adjacent points jumps, the growth will be stopped. The segmentation result of region growing method cannot directly show the whole image. It can display the local area where the seed point is located according to the seed point selected by the operator.

K-Means Clustering Algorithm

K-means clustering algorithm is a clustering-based segmentation algorithm, its idea is to divide the image into k regions by iteration, to ensure that all pixels have the minimum distance to the centre of their generic category. For medical images, k is usually taken as 2, and the gray value is used to replace the measurement index of centre distance. The algorithm process is mainly as follows:

(1) Select two initial clustering points.

(2) The samples are clustered, and the gray difference between all pixels and the central cluster point is calculated to minimize the overall gray difference, which is divided into two types of regions.

(3) The average gray value is calculated for the area divided by clustering, and the average point is taken as the new central clustering point.

(4) Repeat the processes (2) and (3) until the clustering results do not change.

Image Segmentation Experimental Results Comparison

This section uses the above four methods to segment the scene and target, and the results are shown in Figure 15. The original image is used as input in the experiment, due to the epidemic situation, the environment is relatively simple. The original input map adopted in this paper is the chair and apple in the dormitory environment, and the target to be segmented is the apple.

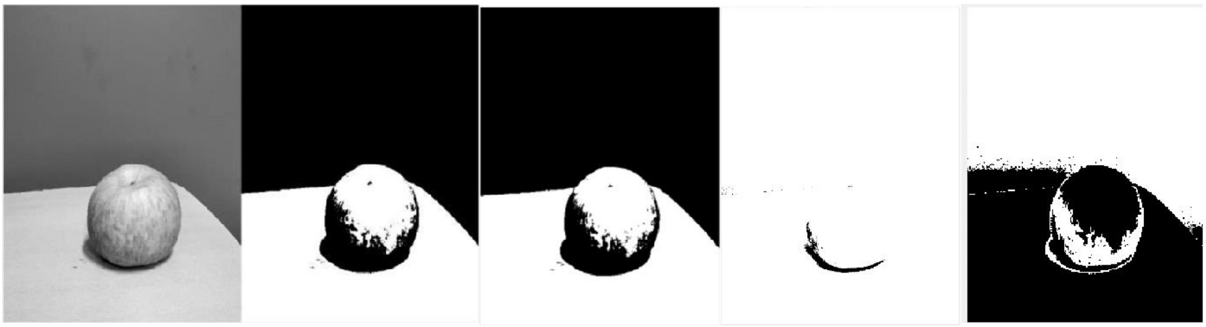

Figure 15 Comparison of Image Segmentation Effects

Due to the poor computing power of the computer, the gray-scale image is used as the input for image data processing in this paper. From left to right, the above five images are the original image, Otsu method, k-means clustering algorithm, one-dimensional maximum entropy algorithm and region growth method respectively. The original input image is shown in Figure 16. Since the chair and the background have a large colour difference, the chair will be converted to white due to its light colour after being converted to a grayscale image, and the background will be converted to black due to its dark colour. In addition, because the texture of the apple in the original image is more obvious, there is also a partial colour difference, which makes it difficult to segment the apple, The segmentation results of k-means clustering algorithm and region growing algorithm can both find that there are obvious segmentation lines at the junction of chair or background and apple, which has achieved good results, but the one dimensional maximum entropy algorithm failed to segment the image, so this paper will further optimize the one dimensional maximum entropy algorithm.

Figure 16 Input Image Original Image

Two-Dimensional Maximum Entropy Algorithm

In the previous section, this paper has introduced the one-dimensional maximum entropy algorithm. The image entropy indicates the average amount of information in the image. Therefore, the basic idea of the maximum entropy algorithm is to maximize the entropy, that is, to maximize the amount of information that the image can present, which also represents the better image segmentation effect. However, it can be seen from Figure 15 that the segmentation effect is poor because there are many meaningless points in the background and chair. Since many experiments have shown that the segmentation effect of two-dimensional maximum entropy algorithm is due to one-dimensional maximum entropy algorithm, this section will mainly introduce two-dimensional gray maximum entropy threshold segmentation method. The two-dimensional maximum entropy algorithm is based on the two-dimensional histogram of the image. Compared with the one-dimensional maximum entropy algorithm, the two-dimensional maximum entropy algorithm introduces the average value of the neighbourhood gray level of each pixel, not only considering a single pixel, but also considering the surrounding distribution of each pixel. In the two-dimensional histogram, the horizontal axis (x-axis) value represents the gray value of the pixel, the vertical axis (y-axis) value represents the neighbourhood gray mean value of each pixel, and the vertical axis (z-axis) value represents the number of pixel points when the gray value is x and the neighborhood gray mean value is y. For one-dimensional maximum entropy algorithm, its core idea is to find the threshold M that maximizes one-dimensional entropy. For the two-dimensional gray maximum entropy algorithm, the key of the algorithm is to find the threshold vector of (i, j) that maximizes the two-dimensional entropy. Figure 17 is a plan view of a two-dimensional histogram, where regions A and B gather pixels of the foreground and background respectively, and regions C and D represent the boundary and noise parts.

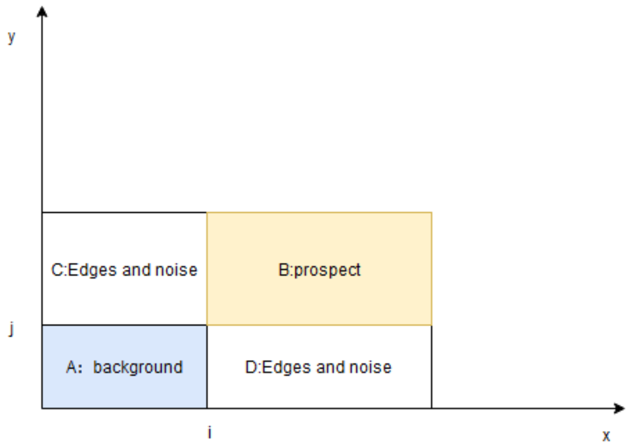

Figure 17 Plan View of Two-Dimensional Histogram

For two-dimensional plan, the probability of each pixel point falling on the (x, y) point is as follows:

$$P_{xy} = \frac{z_{xy}}{M \times N} \tag{7}$$

Where z_{xy} represents the number of pixels when the gray value is x and the average gray value of the neighbourhood is y, $M \times N$ represents the size of the image. The goal of this method is to maximize the two-dimensional gray entropy of the foreground and background. The two-dimensional gray entropy calculation formula is as follows:

$$H = -\sum_{x}\sum_{y} p_{xy} lg p_{xy} \tag{8}$$

The probability of dropping a pixel in the background area A is p_1. The probability of falling in foreground B is p_2, then for $l \times l$ ($k = 256$ in this experiment), then

$$p_1 = \sum_{x=0}^{i-1}\sum_{y=0}^{j-1} p_{xy} \tag{9}$$

$$p_2 = \sum_{x=0}^{I-1}\sum_{y=0}^{I-1} p_{xy} \tag{10}$$

Therefore, the two-dimensional entropy of area A background (H_A) is:

$$\begin{aligned} H_A &= -\sum_{x=0}^{i-j}\sum_{y=0}^{j-i}\left(\frac{p_{xy}}{p_1}\right) lg\left(\frac{p_{xy}}{p_1}\right) = -\left(\frac{1}{p_1}\right)\sum_{x=0}^{i-j}\sum_{y=0}^{j-i}(p_{xy} lg p_{xy} - p_{xy} lg p_1) \\ &= -\left(\frac{1}{p_1}\right) lg p_1 \sum_{x=0}^{i-j}\sum_{y=0}^{j-i} p_{xy} - \left(\frac{1}{p_1}\right) lg p_1 \sum_{x=0}^{i-j}\sum_{y=0}^{j-i} p_{xy} lg p_{xy} \\ &= lg(p_1) + H_1/p_1 \end{aligned} \tag{11}$$

Similarly, the two-dimensional entropy of foreground B area is:

$$H_B = lg(p_2) + H_2/p_2 \tag{12}$$

Where H_1 and H_2 Derived from the above formula, it is:

$$H_1 = -\sum\nolimits_{x=0}^{i-1}\sum\nolimits_{y=0}^{j-1} p_{xy} lg p_{xy} \tag{13}$$

$$H_2 = -\sum_{x=i}^{l-1}\sum_{y=j}^{l-1} p_{xy} lg p_{xy} \tag{14}$$

By the above formula, the sum of the two-dimensional maximum entropy of the foreground and background of the image is a function of the independent variables i and j, as shown below:

$$f(i,j) = H_A + H_B \tag{15}$$

Two-Dimensional Maximum Entropy Threshold Vector Selection

According to the core idea of two-dimensional maximum entropy, the algorithm aims to find a suitable threshold vector (i,j) to maximize two-dimensional entropy $f(i,j)$, and its formula is su:

$$f(i,j) = max\big(f(x,y)\big) \tag{16}$$

Therefore, this paper uses the ergodic method to find the threshold vector (i,j), and uses the double loop structure. Because the gray level of the image selected in this paper is 256 × 256, so both x and y are traversed from the gray value 2 to 255. The algorithm guarantees the comprehensiveness of the results and does not omit any possibility of threshold vectors.

Segmentation Result of Two-Dimensional Maximum Entropy Algorithm

In this paper, the original image in Figure 15 is used as the input. Figure 18 is the experimental result of the two-dimensional gray scale maximum entropy segmentation algorithm. After the calculation of the traversal algorithm, the most appropriate segmentation threshold vector is (125128), while the one-dimensional maximum entropy segmentation threshold in Figure 4.1 is 93. Through comparison, it is not difficult to find that Figure 18 has been greatly improved compared with Figure 15. To some extent, it solves the problem of low segmentation threshold, but compared with the Otsu method, the segmentation effect of k-means clustering algorithm is still poor, and the algorithm is inefficient because it needs 256 * 256 cycles by enumerating.

Figure 18 2D Maximum Entropy Segmentation Result

The Two-Dimensional Gray Gradient Maximum Entropy Algorithm

The 2D data used for maximum entropy of 2D gray level in previous section are pixel gray level and pixel neighbourhood gray level respectively. Considering that the key point of image segmentation is the sensitivity to the boundary, this paper attempts to substitute the image gradient into the gray level index of the replacement pixel neighbourhood. Corresponding to the two-dimensional histogram in Figure 17, the two-dimensional gray gradient maximum entropy algorithm is based on the gray gradient co-occurrence matrix combining gray and gradient information. Similar to previous section, in the gray gradient co-occurrence matrix, the horizontal axis (x-axis) value represents the gray value of the pixel, the vertical axis (y-axis) value represents the gradient value of each pixel, and the vertical axis (z-axis) value represents the number of pixel points when the gray value is x and the gradient value is y. Due to the large gap between the gradient distribution and the corresponding step size of the gray value distribution, in order to improve the data matching of the gray gradient co-occurrence matrix, certain data processing is required first.

Gray Gradient Co-Occurrence Matrix Data Processing

The gray value range of the image is generally 0 to 255, while the gradient value has no defined range. If the original data is used, the corresponding relationship between the x axis and the y axis of the co-occurrence matrix is poor. Therefore, the gradient value is processed in this paper to make it correspond to the gray level distribution from 0 to 255, as shown below:

$$g(x,y) = \frac{g(x_0,y_0)}{255} \times g_{max} \tag{17}$$

In the above formula, $g(x, y)$ represents the gradient value of each pixel after data processing, $g(x_0, y_0)$ represents the gradient value of each pixel before data processing, g_{max} represents the maximum gradient in the image.

The ordinate represents the number of pixel points when the gray value is x and the gradient value is y. The initial image is represented by $z(x_0, y_0)$, which is expressed as:

$$z(x, y) = \frac{z(x_0, y_0)}{\sum_{x_0=0}^{f-1}\sum_{y_0=0}^{g-1} z(x_0, y_0)} \tag{18}$$

In the above formula, $z(x, y)$ represents the z value after data processing, and f and g represent the maximum gray level and maximum gradient level respectively.

Region Description of Gray Gradient Co-Occurrence Matrix

The gray gradient co-occurrence matrix is similar to the two-dimensional histogram in Figure 17 in form and definition, but there are big differences in area division. Its specific image and boundary are shown in Figure 19.

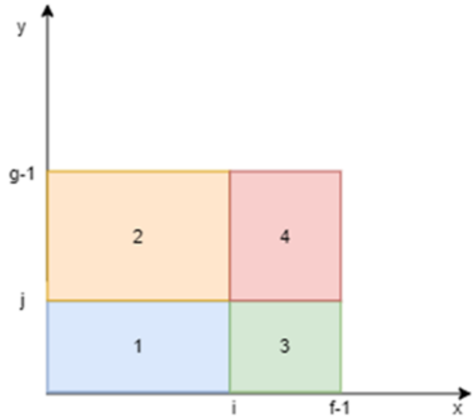

Figure 19 Schematic Diagram of Gray Gradient Co-Occurrence Matrix

The co-occurrence matrix is an $f \times G$ dimension matrix, so its maximum value in the x axis (gray level direction) is $f - 1$, while the maximum value in the vertical axis (gradient direction) is $g - 1$. Similar to 3.1, the optimal threshold vector is defined as (i, j) in this paper. The matrix is divided into four regions of 1,2,3,4 by making a straight line perpendicular to the x axis and y axis through (i, j) points. The main difference from Figure 17 is that since the ordinate changes to a gradient value, the gradient is usually low in the same area, so the objective function of the maximum entropy of the two-bit gray gradient is the sum of the two-dimensional entropy of the first area and the three areas.

$$H(i, j) = H_1 + H_3 \quad (19$$

In this paper, we use the ergodic method to cycle the gray level and gradient from the minimum value to the maximum value to ensure the comprehensiveness of the results.

$$H(i,j) = max\big(H(x,y)\big)$$

Segmentation Result

As shown in Figure 20, the segmentation result of the two-dimensional gray gradient maximum entropy algorithm is better than that of the two-dimensional maximum entropy algorithm. The segmentation threshold is (144153), which is similar to that of the otsu algorithm and k-means clustering algorithm. But the disadvantage of this algorithm is that it traverses data and takes a long time. Previously, through data processing, this paper corresponded the gradient to the gray level one by one, so this paper tried to convert the originally needed gray level gradient vector (i,j) to search (i,i). After many experiments in this paper, the segmentation threshold basically changed little, but the algorithm efficiency has been greatly improved.

Figure 20 2D Gray Gradient Maximum Entropy Segmentation Result

Two Degree of Freedom Manipulator Inverse Kinematics

Through vision odometer, image segmentation and target detection, mobile robot can obtain the position information of itself and the target. Given the robot's end position and posture, solving the position of each motion joint is called the inverse kinematics solution, which is the basis of robot motion planning and trajectory control.

Inverse Solution Process

As shown in Figure 21, this is a simplified model of a two degree of freedom mechanical arm. We take the starting point O of the mechanical arm as the coordinate origin. The length of the starting mechanical arm is l_1, and the length of the end mechanical arm is l_2. The starting mechanical arm forms an angle θ_1 counterclockwise with the horizontal direction, and the end mechanical arm forms an angle θ_2 counterclockwise with the starting mechanical arm. Since the mechanical arm is a two degree of freedom mechanical arm, we can set the target coordinate as (x, y) by adjusting the angle $\theta 1$ and angle θ_2, The formula is as follows:

$$\begin{cases} l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) = x \\ l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) = y \end{cases} \tag{20}$$

After angle θ_1 and angle θ_2 are solved, each mechanical arm is controlled by the motor to turn to the corresponding angle, and the two degree of freedom mechanical arm can grasp the target object.

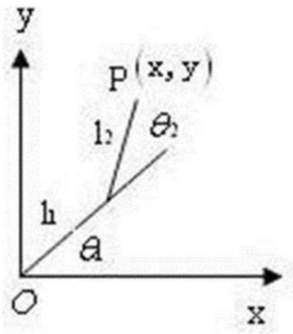

Figure 21 Mathematical Model of Two Degree of Freedom Manipulator

Motion Space of Two Degree of Freedom Manipulator

In theory, angle 1 and angle 2 can be any angle within the range of 0 to 360 degrees, so their ideal motion space is a circle with radius of l1 and the sum of the two. However, in practice, due to the length and thickness of each mechanical arm cannot be ignored, and the rotation will also encounter obstacles, the actual workspace is far less than the theoretical workspace.

Conclusions

The main research object of this paper is a mobile robot. Through a simple grab task moving to a designated location as a requirement, the visual technology that needs to be used in this task is simulated. Due to the epidemic situation and time constraints, this paper only analyses the visual technology that needs to be used in a small part of the tasks. This paper first describes the hardware architecture of the mobile robot, specifically describes the sensor system technology, and describes the method of acquiring depth information with monocular cameras and binocular cameras, which makes up for the loss of depth dimension information in image information. Then this paper describes the visual odometer part of slam, and carries out feature point matching, odometer estimation part of the code simulation. After that, this paper describes the knowledge of image segmentation, and uses Otsu method, k-means clustering, region growth, and one-dimensional maximum entropy algorithm to perform scene segmentation simulation respectively. Due to the poor segmentation effect of one-dimensional maximum entropy algorithm, this paper introduces two-dimensional gray gradient maximum entropy algorithm for optimization and achieves good segmentation effect. Finally, this paper describes the most basic inverse kinematics of the two degrees of freedom manipulator, and how to choose the appropriate joint angle to grasp the target when the target coordinates and the initial coordinates of the manipulator are known. However, because the conventional scene segmentation can only segment the foreground and background, the segmentation effect still has some limitations. In the future, we can try to use the depth learning method to better segment the object contour and identify the target object.

Reference

Arrais, R., Oliveira, M., Toscano, C., & Veiga, G. (2017). A mobile robot based sensing approach for assessing spatial inconsistencies of a logistic system. *Journal of Manufacturing Systems*, *43*, 129-138. https://doi.org/https://doi.org/10.1016/j.jmsy.2017.02.016

Ben Azouz, A., Esmonde, H., Corcoran, B., & O'Callaghan, E. (2015). Development of a teat sensing system for robotic milking by combining thermal imaging and stereovision technique. *Computers and Electronics in Agriculture*, *110*, 162-170. https://doi.org/https://doi.org/10.1016/j.compag.2014.11.004

Choudhary, P. K., Routray, S., Upadhyay, P., & Pani, A. K. (2022). Adoption of enterprise mobile systems – An alternative theoretical perspective. *International Journal of Information Management*, *67*, 102539. https://doi.org/https://doi.org/10.1016/j.ijinfomgt.2022.102539

Geng, Y., Yuan, M., Tang, H., Wang, Y., Wei, Z., Lin, B., & Zhuang, W. (2022). Robot-based mobile sensing system for high-resolution indoor temperature monitoring. *Automation in Construction*, *142*, 104477. https://doi.org/https://doi.org/10.1016/j.autcon.2022.104477

Houtman, W., Lopez Martinez, C. A., Wang, S., Ketels, A., Bruyninckx, H. P. J., & van de Molengraft, M. J. G. (2021). Dynamic control of steerable wheeled mobile platforms applied to an eight-wheeled RoboCup Middle Size League soccer robot. *Mechatronics*, *80*, 102693. https://doi.org/https://doi.org/10.1016/j.mechatronics.2021.102693

Li, H., Huang, F., & Chen, Z. (2022). Virtual-reality-based online simulator design with a virtual simulation system for the docking of unmanned underwater vehicle. *Ocean Engineering*, *266*, 112780. https://doi.org/https://doi.org/10.1016/j.oceaneng.2022.112780

Lin, T. Y., Shi, G., Yang, C., Zhang, Y., Wang, J., Jia, Z., Guo, L., Xiao, Y., Wei, Z., & Lan, S. (2021). Efficient container virtualization-based digital twin simulation of smart industrial systems. *Journal of Cleaner Production*, *281*, 124443. https://doi.org/https://doi.org/10.1016/j.jclepro.2020.124443

Luo, X., Mu, D., Wang, Z., Ning, P., & Hua, C. (2023). Adaptive full-state constrained tracking control for mobile robotic system with unknown dead-zone input. *Neurocomputing*, *524*, 31-42. https://doi.org/https://doi.org/10.1016/j.neucom.2022.12.025

Nwachioma, C., & Pérez-Cruz, J. H. (2021). Analysis of a new chaotic system, electronic realization and use in navigation of differential drive mobile robot. *Chaos, Solitons & Fractals*, *144*, 110684. https://doi.org/https://doi.org/10.1016/j.chaos.2021.110684

Shafaei, S. M., & Mousazadeh, H. (2023). Experimental comparison of locomotion system performance of ground mobile robots in agricultural drawbar works. *Smart*

Agricultural Technology, *3*, 100131. https://doi.org/https://doi.org/10.1016/j.atech.2022.100131

Suwoyo, H., Abdurohman, A., Li, Y., Adriansyah, A., Tian, Y., & Ibnu Hajar, M. H. (2022). The Role of Block Particles Swarm Optimization to Enhance The PID-WFR Algorithm. *International Journal of Engineering Continuity*, *1*(1), 9-23. https://doi.org/10.58291/ijec.v1i1.37

Suwoyo, H., & Harris Kristanto, F. (2022). Performance of a Wall-Following Robot Controlled by a PID-BA using Bat Algorithm Approach. *International Journal of Engineering Continuity*, *1*(1), 56-71. https://doi.org/10.58291/ijec.v1i1.39

Teck, S., & Dewil, R. (2022). Optimization models for scheduling operations in robotic mobile fulfillment systems. *Applied Mathematical Modelling*, *111*, 270-287. https://doi.org/https://doi.org/10.1016/j.apm.2022.06.036

Tengilimoglu, O., Carsten, O., & Wadud, Z. (2023). Implications of automated vehicles for physical road environment: A comprehensive review. *Transportation Research Part E: Logistics and Transportation Review*, *169*, 102989. https://doi.org/https://doi.org/10.1016/j.tre.2022.102989

Xue, K., Wang, Z., Shen, J., Hu, S., Zhen, Y., Liu, J., Wu, D., & Yang, H. (2021). Robotic seam tracking system based on vision sensing and human-machine interaction for multi-pass MAG welding. *Journal of Manufacturing Processes*, *63*, 48-59. https://doi.org/https://doi.org/10.1016/j.jmapro.2020.02.026