S & M 2794

Application of Path Planning and Image Processing for Rescue Robots

Chuan-Yen Lu, Chia-Chia Kao, Yu-Hsiang Lu, and Jih-Gau Juang*

Department of Communications, Navigation and Control Engineering, National Taiwan Ocean University,
2 Pei-Ning Road, Keelung 20224, Taiwan

(Received April 5, 2021; accepted November 16, 2021)

Keywords: A* algorithm, path planning, image processing, WMR, laser sensor, DGPS

The objective of this study was to integrate different position sensors and a camera in a wheeled mobile robot (WMR) for use in a rescue task. A differential global positioning system (DGPS) was used to locate the WMR position and the goal position. A laser sensor was utilized to detect the locations of obstacles and edge values. The edge values were applied to the A* algorithm to search for the shortest path for the WMR. While searching for a target, a webcam was used to capture images of the surroundings. Color segmentation was achieved and the desired color was captured. The captured color target was then used for tracking control of the rescue task. The A* algorithm, together with image processing and laser sensors, was able to obtain the shortest path to the desired color target in an unknown environment. Obstacle avoidance control was based on a fuzzy system. Experiments confirmed that the proposed control scheme worked properly and was suitable for rescue tasks.

1. Introduction

In recent years, researchers have paid increasing attention to improving robotic systems, and many related applications and technologies have been widely developed. In particular, robots have begun to be used instead of human beings for some extremely dangerous tasks. Therefore, robot navigation and obstacle avoidance are two major issues that have been widely discussed. Path planning, obstacle avoidance, parking control, object searching and tracking, the cooperation of multiple robots, and so forth have been applied to the design of wheeled mobile robot (WMR) control schemes. Zhao and BeMent addressed six common kinds of three-wheeled robot models and their ability of control.(1) Leow *et al.* analyzed the kinetic model of an all-direction wheeled robot.(2) Chung *et al.* utilized two wheels with different speeds for position control.(3) Wang and Juang utilized a localization system for the path planning and parking control of a WMR.(4) Lin applied an omnidirectional camera and laser range finder to extract point features and line features as landmarks.(5) Wang used a low-end camera for simultaneous localization and mapping (SLAM), which had the advantages of standardized hardware and software interfaces with a low price, as well as the potential for wide application in robotic

*Corresponding author: e-mail: jgjuang@ntou.edu.tw

https://doi.org/10.18494/SAM.2022.3546

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

systems.(6) In these studies, feature points were obtained by a complex and time-consuming process. However, laser sensors can be used to obtain environmental data. Martinez *et al.* used a laser scanner for following mobile objects and avoiding obstacles.(7) Yamakawa *et al.* proposed two fixed-angle laser scanners that could reconstruct the 3D shape of an obstacle, and the reconstructed shapes were used to produce a potential field map for mobile robot path planning.(8) Chen used the data of a laser range finder to develop a feature map. The feature map and corresponding preset waypoints were applied to calculate the SLAM algorithm for a moving robot in an unknown environment.(9) Much research has been devoted to the problem of vehicle path planning using, for example, topological methods, the Dijkstra algorithm, the A* algorithm, the B-spline, genetic algorithms, and neural networks.(10,11) Merchan-Cruz and Morris proposed a novel fuzzy genetic algorithm approach to tackle the problem of trajectory planning for two collaborative robot manipulators sharing a common workspace.(12) Zhong *et al.* presented a new methodology based on a neural network for robot path planning, in which an improved Hopfield-type neural network model was established for propagating the target activity among neurons in the manner of physical heat conduction, which guaranteed that the target and obstacles remained at the peak and the bottom of the activity landscape of the neural network, respectively.(13)

Fuzzy logic has the advantage of the solution to a problem being cast in terms that human operators can understand. Fuzzy logic has the benefit of better implementation, greater fault tolerance, and more fitting usage on nonlinear systems. Shi *et al.* installed multiple sensors on a WMR and utilized fuzzy reasoning values to control it.(14) Lee *et al.* proposed rapid path planning and fuzzy logic control to force a WMR to trace the desired path.(15,16) Regarding real-time obstacle avoidance, the majority of studies have used fuzzy controllers to control WMRs.(17) In the case of outdoor navigation, the global positioning system (GPS) has been widely used for mobile robots.(18) Hamid *et al.* proposed a mobile robot equipped with a GPS navigation system and an obstacle avoidance system that utilized a low-cost mobile structure, a GPS module, and a sonar sensor.(19) Kim *et al.* used differential GPS and odometry data within the framework of an extended Kalman filter to localize a mobile robot. They also proposed an algorithm to detect curbs through a laser range finder.(20) In the target image recognition part, the vision sensor helped with searching and recognition. Many studies on WMRs have used color image recognition for robotic mobile control.(21) Ravari *et al.* proposed a vision-based algorithm for mobile robot navigation in unknown outdoor environments.(22) Tsai proposed color filter array interpolation for an image sensor and the visual tracking control design of a WMR.(23) The focus of automatic guidance vehicle development is on how to make these vehicles work effectively in various environments. Many studies of intelligent robots have been applied in the areas of home services, health care, space exploration, the military, and entertainment.(24,25) In our previous work, GPS and image processing were successfully applied to outdoor patrols.(26) In this study, we integrated a laser sensor, a vision sensor, a differential global positioning system (DGPS), image processing, A* algorithm path planning, and a fuzzy system into a WMR. The WMR could move along the shortest path and avoid obstacles to reach the desired goal position and utilized a vision sensor to search for the target of the rescue task.

2. System Description

The experiment on the proposed control scheme was performed using an Ihomer WMR, as shown in Fig. 1. The WMR had a length of 480 mm, a width of 455 mm, and a height of 250 mm. The Ihomer WMR is a two-wheeled differentially driven car-like robot with two small casters located at its rear and front. Two 18 V DC motors were installed to provide torque (7.2 mNm), and each motor controlled one wheel. The maximum speed of the robot was 1.6 m/s and its total weight was 15.2 kg. A Sick LMS100 laser sensor was placed on top of the WMR, as shown in Fig. 1. Its field of view was 270°, the scanning range was 20 m, the resolution of the angular step width was 0.5°, and the rotation frequency was 50 Hz. Ethernet was used to connect the LMS100 sensor and the on-board computer.

The LMS100 sensor is an electro-optical laser measurement sensor that electrosensitively scans the perimeter of its surroundings in a plane with the aid of laser beams. It measures its surroundings using 2D polar coordinates. If a laser beam is incident on an object, the position can be determined in the form of distance and direction, as shown in Fig. 2.(27) The LMS100 sensor emits pulsed laser beams using a laser diode. If such a laser pulse is incident on an object or person, it will be reflected at its surface. The reflection is detected in the sensor's receiver using a photodiode. The distance to the object is calculated from the propagation time that the light requires from emission to reception of the reflection at the sensor. A similar principle of pulse propagation time measurement is used by radar systems. Figure 3 shows the network camera (Microsoft LifeCam Studio 1080P Full-HD) used in this study to identify the color of the target. The camera was installed on top of the WMR.

A DGPS typically uses measurements from two or more remote receivers to calculate the differences (corrections) between measurements and thus provide more accurate position solutions. With a DGPS, it is possible to monitor variations in the GPS signal and communicate these variations to the other receiver. The second receiver can then correct its calculations for better accuracy. A DGPS relies on the concept that errors in the position at one location are similar to those for all locations within a given area. The reference station collects the range measurements from each GPS satellite in view and forms the differences (corrections) between the calculated distances to the satellites and the measured pseudo-ranges to the satellites. These corrections are then used to satisfy the Radio Technical Commission for Maritime Services (RTCM) standard established for transmitting differential corrections and are broadcast to the remote receivers using a data communication link. Each remote receiver applies the transmitted DGPS corrections to its range measurements of the same satellites. By this technique, spatially correlated errors such as satellite orbital errors, ionosphere errors, and tropospheric errors can be significantly reduced, while common errors such as clock bias can be eliminated, thus improving the position accuracy of the GPS. A GMS-2 DGPS receiver was installed on top of the WMR at the center position, as shown in Fig. 4. Through Wi-Fi communication with the GPS receiver, real-time GPS information in the RTCM format can be obtained. The differential results were calculated using the GMS-2 receiver. The GMS-2 receiver transmitted the location and direction data to the computer via USB and RS232 connections. The GMS-2 receiver was a single-frequency GPS+GLONASS L1 DGPS receiver. The differential results were calculated using the

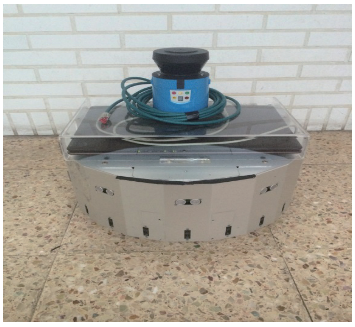


Fig. 1. (Color online) LMS100 sensor on top of the Ihomer WMR.

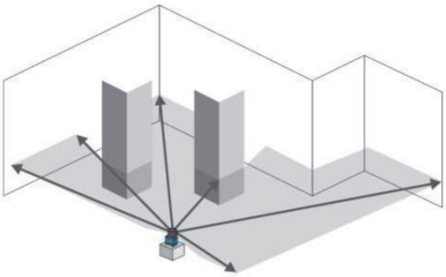
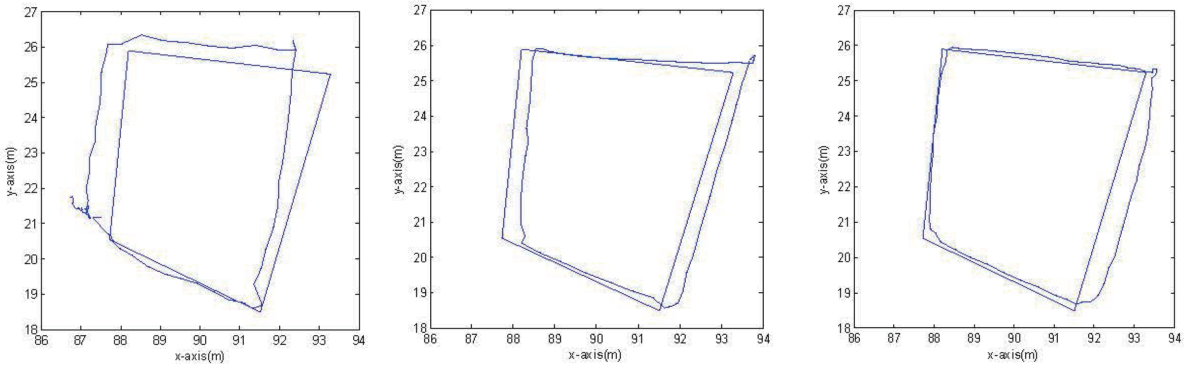


Fig. 2. Scanning simulation of the LMS100 sensor.

Fig. 3. (Color online) Microsoft webcam.

Fig. 4. (Color online) GMS-2 DGPS receiver.



(a) (b) (c)

Fig. 5. (Color online) Dynamic average error of the GMS-2 receiver in DGPS mode with a moving velocity of 0.5 m/s. (a) 0.7431, (b) 0.7016, and (c) 0.6347 m.

GMS-2 receiver in the National Marine Electronics Association (NMEA) format. The dynamic average error was measured three times along a rectangular path and was compared with the static measurement data, as shown in Fig. 5.

3. Image Processing

The webcam captured the image in the RGB color space; then, the RGB image was transformed to the HSL color space. Color segmentation was used to obtain the desired color(28) and then binarize the color of the target. A filter was used to eliminate noise during image processing. Image thresholding consisted of changing the image brightness to 0 or 255, where 0 represented black and 255 represented white.(29) The original image could therefore be simplified into the foreground and background. The foreground represented the captured target, while the background was the image to be filtered out. This method reduced the amount of image data processing. When thresholding an image, finding an appropriate threshold is very important as the threshold will affect the captured image of the target. The image thresholding was based on comparing the pixel value at each point in the image with the threshold value. If the pixel value was larger than the threshold value, it was set to 255; if the pixel value was smaller than the threshold value, it was set to 0. In image processing, it is necessary to perform noise reduction before target detection. The mean filter is a linear filtering technology(30) that can be used to remove Gaussian noise. The principal neighborhood average method uses the mean value of each pixel in a certain image region to replace each pixel of the original image. The median filter is a nonlinear filtering technology that can be used to remove noise from images and signals. It is particularly useful for speckle noise and salt-and-pepper noise. It is also useful for detecting instances of undesirable edge blur to save edge features. In this procedure, it is necessary to first set an $n \times n$ mask, where n must be odd. Then, a value is extracted to replace the original image value on an image after sorting in the mask.

3.1 Color segmentation

Color segmentation can recognize all colors from an image and extract specific colors. To extract specific colors, the color spectrum is used to decide the thresholds for the hue, saturation, and lightness.(31) The webcam in a detection system then only detects the desired color. We used this process to detect the color of the target. An example is shown in Fig. 6, which shows different blocks in different colors. We located the green color by setting the hue value at 120°, the saturation value at 100%, and the lightness value at 0.5. In Fig. 7, the red frame shows the detected section, and only the green square was detected.

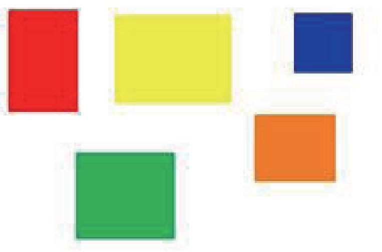


Fig. 6. (Color online) Blocks of different colors.

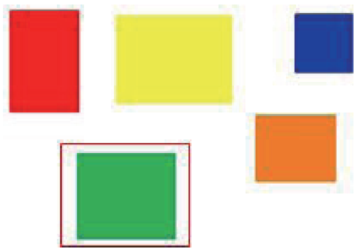


Fig. 7. (Color online) Detected section and green square.

3.2 Distance estimation

The real-world distance was computed by a geometric method, which was used to calculate the distance between the WMR and the target. Three prerequisite angles were calculated:

$$\alpha = \tan^{-1}\left(\frac{E}{b}\right), \tag{1}$$

$$\theta = \tan^{-1}\left(\frac{E}{L_y}\right), \tag{2}$$

$$\beta = \tan^{-1}\left(\frac{L_x / 2}{L_y + b}\right), \tag{3}$$

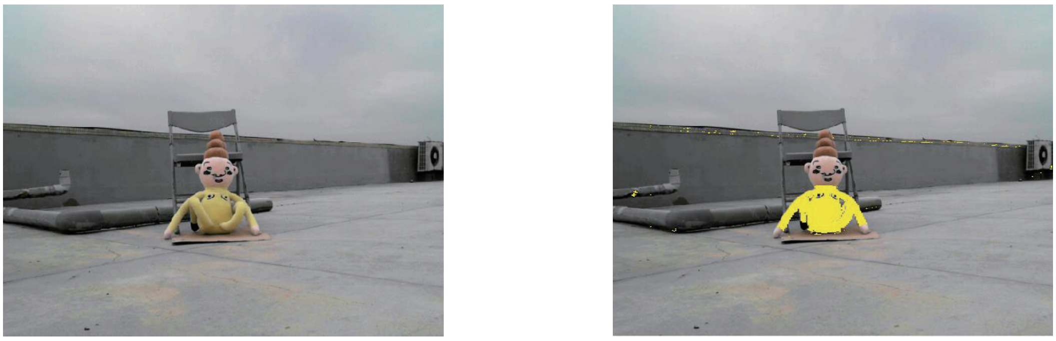
where L_x is the largest width that the camera can see and is the length from the left to the right of the view, E is the height from the camera to the floor, b is the distance from the projection point of the camera on the floor to the closest viewpoint, and L_y is the largest range that the camera can see. Two prerequisite positions were calculated using Eqs. (4) and (5),(32) in which the real positions are X and Y, μ and v are the horizontal and vertical pixels of the center coordinates of the target, respectively, and H and W represent the pixel lengths of the image provided by the camera. We obtained the center coordinates of the target after the image processing. The estimated distance was then used to calculate the distance between the WMR and the target.

$$Y = \frac{E}{\tan\left[(\alpha - \theta)\frac{v}{W}\right]} \tag{4}$$

$$X = \tan\left[\beta\left(\frac{2\mu}{H} - 1\right)\right] \cdot Y \tag{5}$$

3.3 Pre-processing experiment results

The RGB image was first transformed to the HSL color space, and then the color segmentation process was applied in the HSL color space. We intended the robot to search for yellow features with threshold values of $25 < H < 46$, $47 < S < 255$, and $0 < L < 247$, as shown in Fig. 8. Color features were used to search for the target because when the WMR searched for or approached a target, the image of the target may have been distorted. The use of features requires fewer computations and provides more robust performance, making them suitable for real-time control. After color segmentation and image binarization, there was speckle noise, as shown in Fig. 9. We used the median filter to filter out the noise, as shown in Fig. 10. Next, we set the white frame as the detection range. We captured the desired target by setting the area pixel threshold, as shown



(a) (b)

Fig. 8. (Color online) (a) RGB image and (b) HSL image.

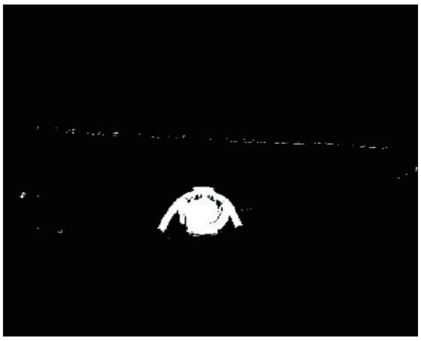


Fig. 9. Speckle noise in the binary image.

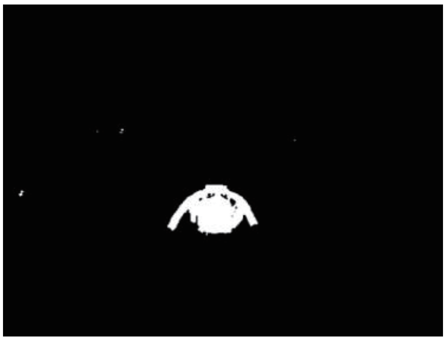
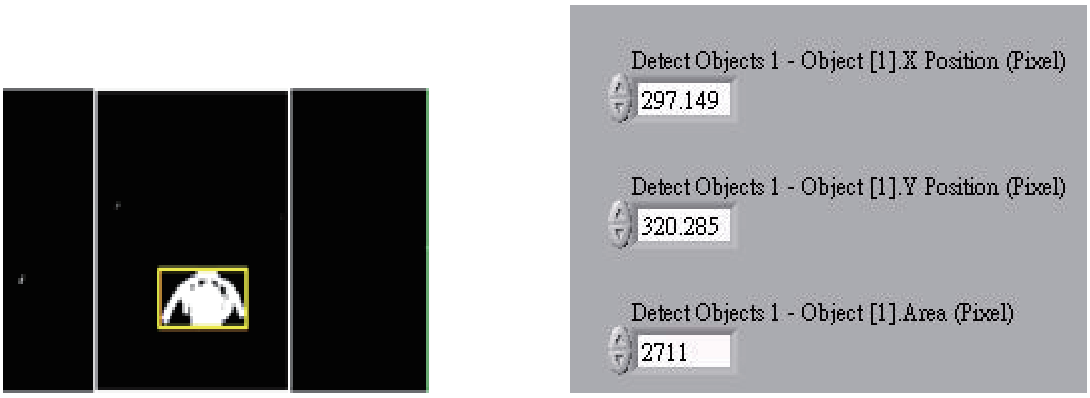


Fig. 10. Median filter.



Detect Objects 1 - Object [1].X Position (Pixel)
297.149

Detect Objects 1 - Object [1].Y Position (Pixel)
320.285

Detect Objects 1 - Object [1].Area (Pixel)
2711

(a) (b)

Fig. 11. (Color online) (a) Detection range shown by white frame. The target is captured in the yellow box. (b) Center position of target and pixel area information.

in Fig. 11(a). We then obtained the target's center position (the *x* and *y* pixels) and the pixel area of the target, as shown in Fig. 11(b). This information was then used to calculate the distance between the WMR and the target.

4. Control Scheme

The flow chart of the control sequence is shown in Fig. 12. In this study, we used a laser sensor to detect the locations of obstacles and the edge values. We used the edge values with the A* algorithm to search for the shortest path and used fuzzy theory to safely move along the shortest path. We then used a DGPS to locate the WMR position and the goal position to search for the desired target in an unknown environment.

4.1 A* path planning algorithm

The A* algorithm is based on the Dijkstra algorithm and the best-first search algorithm.(33) Its basic frame is based on a graph traversal search strategy, but it uses the heuristic function to estimate the cost from a node to the destination, which then guides the search toward the destination. The heuristic function is expressed as

$$f^* = g(j) + h^*(j), \tag{6}$$

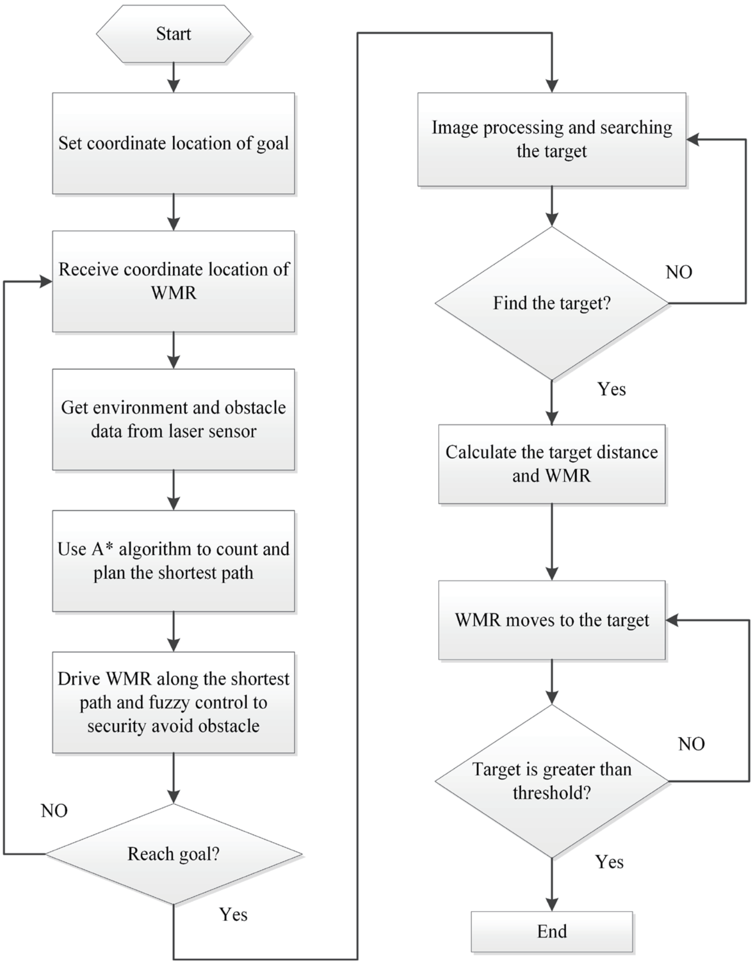

Start
Set coordinate location of goal
Receive coordinate location of WMR
Get environment and obstacle data from laser sensor
Use A* algorithm to count and plan the shortest path
Drive WMR along the shortest path and fuzzy control to security avoid obstacle
NO
Reach goal?
Yes
Image processing and searching the target
Find the target?
NO
Yes
Calculate the target distance and WMR
WMR moves to the target
Target is greater than threshold?
NO
Yes
End

Fig. 12. Flow chart of the control sequence.

where $g(j)$ is the actual cost from the origin node to node j (i.e., the cost of finding the optimal path), $h^*(j)$ is the estimation of the minimum cost from node j to the destination node, which depends on the heuristic information of the problem area, and $f^*(j)$ is the estimation of the minimum travel cost among all routes from the origin node to the destination node. If $h^*(j) = 0$, there is no heuristic information and the Dijkstra algorithm must be used. Users can choose an appropriate $h^*(j)$ for different cases, but $h^*(j)$ must obey the following consistency rule: $h^*(j)$ must be less than $h(j)$, which is the actual cost from node j to the destination node. If the cost function accords with the rule, then the problem has optimal results and the A* algorithm will find the optimal route. The advantage of the A* algorithm is that it guides the search toward the destination node using the heuristic function, which can reduce the search depth and the number of nodes in the network and allow it to consume less memory.(37)

The search process of the A* algorithm is as follows:(34)

Step 1 Mark the initial node and join it to the unmarked subsequent node or the child node.

Step 2 Calculate the heuristic function value for each subsequent node, sort these values in numerical order according to the value of the heuristic function, and identify and mark the node with the minimum heuristic function value. The search is continued until the current node becomes the target node.

Step 3 Record the shortest path if the current node is not the target node.

The definition of the heuristic function is the key point to the A* algorithm. Expressed as Eq. (7), it starts with the idea of finding a minimum cost path by intuitively using the Euclidean distance heuristic. The cost function $g(j)$ is also calculated using the Euclidean distance, as shown in Eq. (8). A path diagram with the cost of nodes is shown in Fig. 13.

$$h^*(j) = \sqrt{\left(x_j - x_{goal}\right)^2 + \left(y_j - y_{goal}\right)^2} \tag{7}$$

$$g(j) = \sqrt{\left(x_j - x_c\right)^2 + \left(y_j - y_c\right)^2} \tag{8}$$

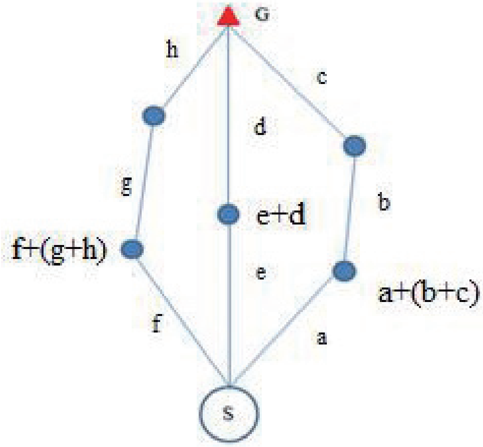


Fig. 13. (Color online) Path diagram with the cost of nodes.

Here, (x_j, y_j) is the position of node j, (x_{goal}, y_{goal}) is the location of the goal point, and (x_c, y_c) is the position of the initial point. Because our route searching method is used for real-time dynamic WMR control, the initial point is simply the current position of the WMR. The route searching algorithm is only applied when there are obstacles in the direction from the WMR to the goal; without any obstacles, the optimal path will be the direct route. When the WMR moves to the first node of the shortest path, it will continue to calculate the cost of other nodes and then move to the second node of the shortest path. The WMR will continue to move and plan the path iteratively until it reaches the desired goal point.

Although the A^* algorithm method can quickly achieve path planning, there are two cases of the estimation function $h^*(j)$ that will affect the optimal path solution.(35)

Case 1: If $h^*(j)$ is less than or equal to the actual distance from node j to the target node, it will consider more search points and have a larger search range and low efficiency, but it will also reach the optimal path solution.

Case 2: If $h^*(j)$ is greater than the actual distance from node j to the target node, it will consider fewer search points and have a smaller search range and higher efficiency, but it cannot be guaranteed to reach the optimal path solution.

We considered the above two cases and improved $h^*(j)$ to ensure that the efficiency was higher and that it found the optimal path solution. Another equation, $P(path)$, was introduced to improve $h^*(j)$. $P(path)$ was defined as the path cost, expressed as(35)

$$P(path) = w_t t_{path} + w_c P_{dis} , \tag{9}$$

where t_{path} is the required time to traverse the path, P_{dis} is the cost of the path distance, and the weight factors w_t and w_c satisfy $w_t + w_c = 1$. Replacing $h^*(j)$ with $P(path)$ in Eq. (6) improved the estimation function. Combining Eq. (8) with Eq. (9) resulted in the following new cost function:

$$f^*(i,j) = g(i,j) + w_t t(i,j) + w_c p_{dis}(i,j) , \tag{10}$$

where $g(i, j)$ is the distance from node i to node j, $t(i, j)$ is the time required by node i to transverse the path to node j, and $P_{dis}(i, j)$ is the distance from node i to node j. The improved estimation function reduces the search time and improves the search efficiency; thus, it was used in the navigation of the actual path.

4.2 Fuzzy controller for obstacle avoidance

For collision avoidance, it was necessary to consider how to move the WMR safely along the shortest path at a constant speed. The minimum safe distance between the WMR and obstacle was set at 30 cm. An LMS100 laser sensor was placed on the WMR as shown in Fig. 14, where the 45° angle of the laser sensor on the WMR is marked as LR, the measured distance is marked as dr, the 90° angle is marked as LF, the measured distance is marked as df, the 135° angle is marked as LL, and the measured distance is marked as dl.

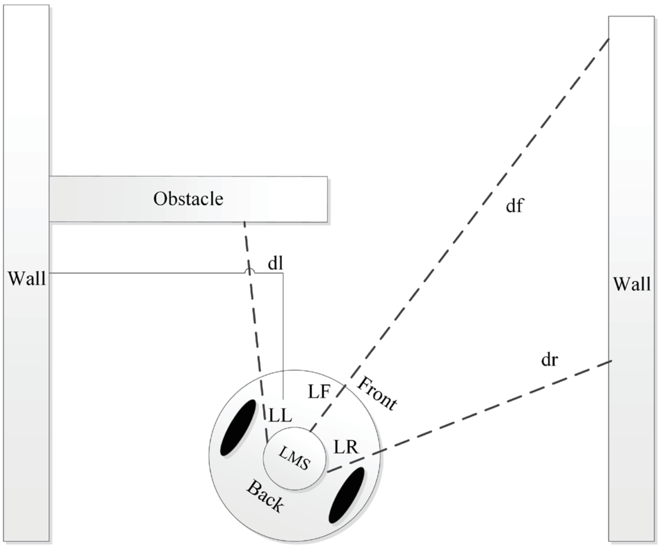


Fig. 14. Input variables of the WMR from the LMS100 sensor.

After using the A* algorithm to find the shortest path node, the WMR required avoidance control to move safely along the shortest path. Fuzzy theory was applied to avoid obstacles on the path.(36) The fuzzy control structure had three inputs and one output. The signals from 45°, 90°, and 135° relative to the LMS100 sensor were the inputs, and the signal represented the distance. The fuzzy sets of the inputs were set as F (far), M (medium), and N (near). Output A was the angle for turning right or left, and the fuzzy sets of the output were RB, RS, Z, LS, and LB. The fuzzy rules were as follows:

Rule 1: If LL is F, LF is F, and LR is F, then A is Z.
Rule 2: If LL is M, LF is F, and LR is F, then A is Z.
Rule 3: If LL is N, LF is F, and LR is F, then A is RS.
Rule 4: If LL is N, LF is N, and LR is F, then A is RB.
Rule 5: If LL is N, LF is N, and LR is M, then A is RB.
Rule 6: If LL is N, LF is M, and LR is M, then A is RS.
⋮
Rule 26: If LL is F, FL is M, and RL is N, then A is LS.
Rule 27: If LL is F, FL is N, and RL is N, then A is LB.

5. Experimental Results

The LMS100 laser sensor used in the experiment had a scanning angle from −45° to 255°. Therefore, the maximum effective surrounding angle of the robot set in the simulation was 270° and the minimum safe distance was 50 cm. Figure 15 shows the simulation results when the A* algorithm was used to find the shortest nodes and fuzzy obstacle avoidance was used to reach the goal point in different environments, which were labeled as Map-1, Map-2, and Map-3.

Figure 16 shows the simulation results obtained using the improved A* algorithm to find the shortest path. A virtual wall was added to the edge value to represent a safe distance, as shown

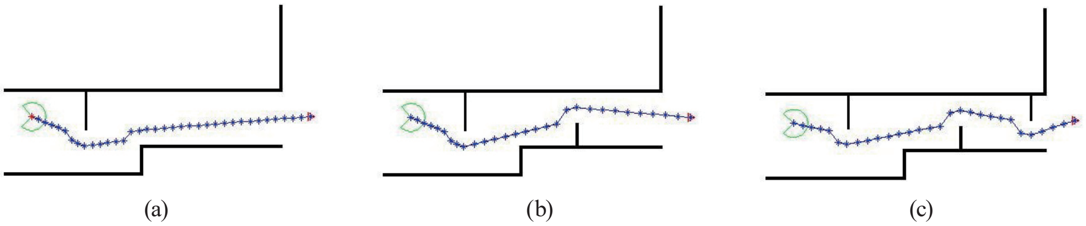


Fig. 15. (Color online) Effective surrounding angle of the robot: (a) Map-1, (b) Map-2, and (c) Map-3.

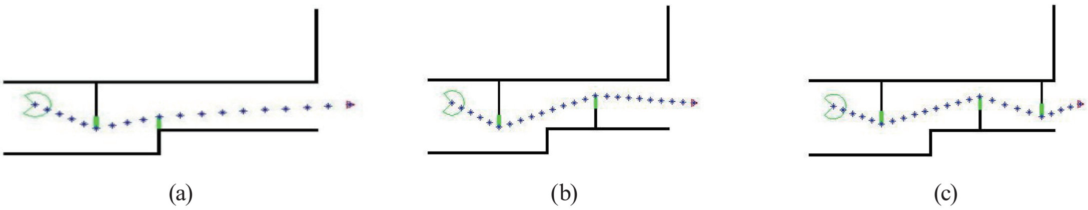


Fig. 16. (Color online) Simulation results when the virtual safe distance was set to 60 cm: (a) Map-1, (b) Map-2, and (c) Map-3.

Table 1
Total path lengths in cm for A* algorithm and improved A* algorithm.

Map	A* algorithm	Improved A* algorithm
Map-1	1.0801×10^3	1.0324×10^3
Map-2	1.0893×10^3	1.0459×10^3
Map-3	1.0903×10^3	1.0474×10^3

by the green wall in the figure. A comparison of the total path lengths found using the A* algorithm and improved A* algorithms is shown in Table 1. The path length found using the improved A* algorithm was shorter than that found using the A* algorithm.

A goal point was set at N25.14981472438608 and E121.77759770036515 in the initial setup of Map-3. The control scheme and image searching in the experiment consisted of path finding using the A* algorithm, safe moving, and target searching. Figure 17 shows the WMR moving along the shortest path, avoiding an obstacle, and reaching the goal in the Map-3 environment. After reaching the goal, the WMR searched for the desired target (which was colored yellow) and moved to the target.

The WMR starts at a starting point and uses the environmental data from the laser sensor [Fig. 17(a)], then finds the first node of the shortest path [Fig. 17(b)]. The WMR moves along the shortest path to the first node and maintains a safe distance from the obstacles [Fig. 17(c)], then reaches the turning point, after which it turns to the front line and obtains the second cluster from the laser sensor [Fig. 17(d)]. The WMR finds the second node of the shortest path [Fig. 17(e)], then moves along the shortest path to the second node and maintains a safe distance from the obstacles [Fig. 17(f)]. The WMR reaches the second turning point, turns to the front line to obtain the third cluster from the laser sensor [Fig. 17(g)], and then finds the third node of the

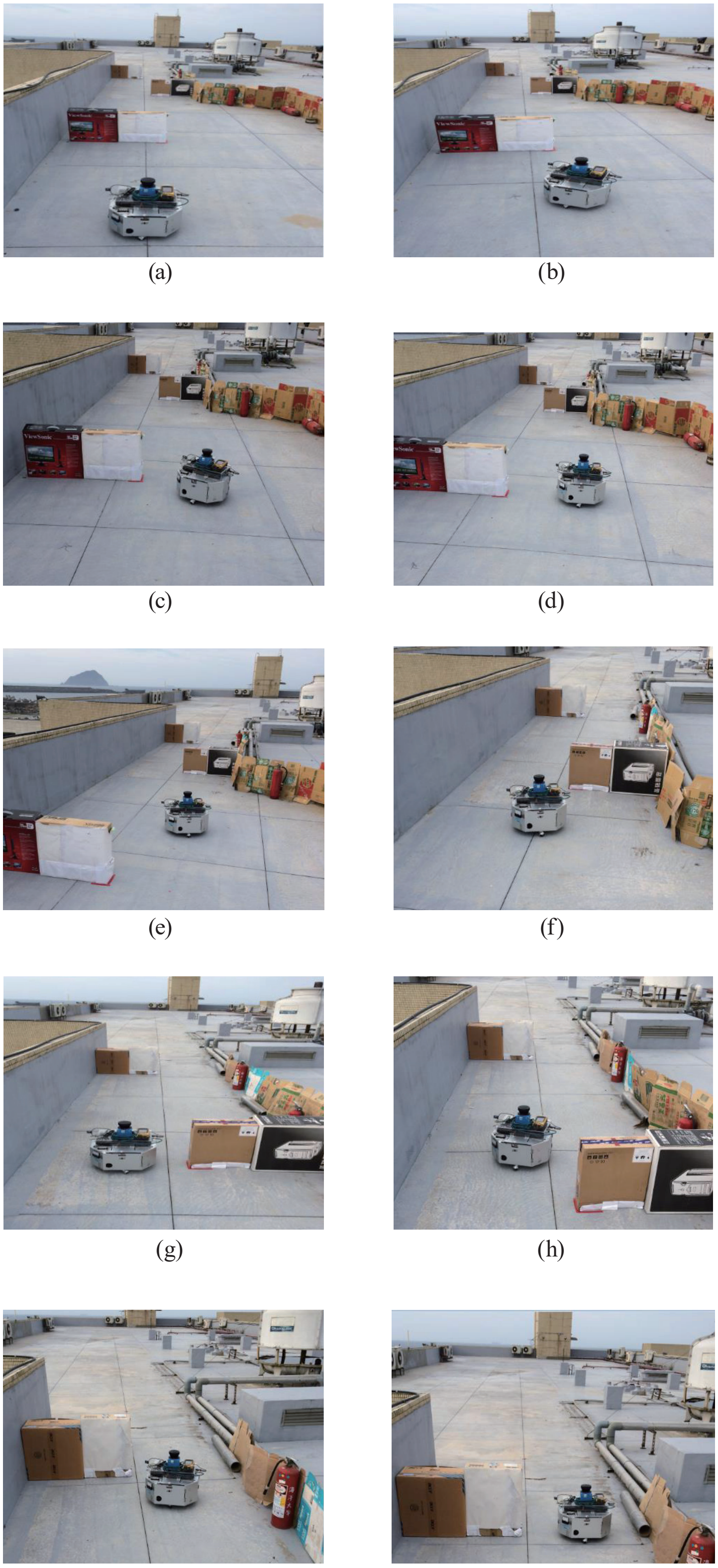
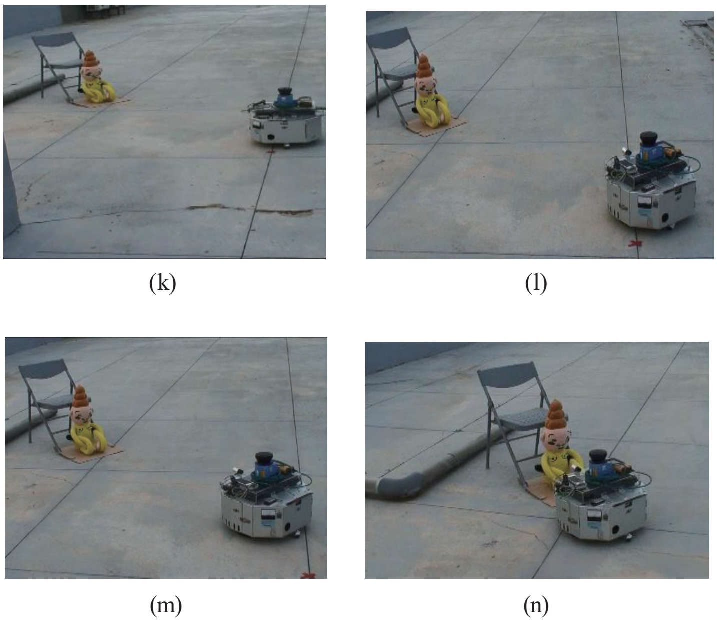


Fig. 17. (Color online) WMR moving along the shortest path, avoiding obstacles, reaching the goal, then finding the desired target in the Map-3 environment.



(k) (l)

(m) (n)

Fig. 17. (Color online) (Continued) WMR moving along the shortest path, avoiding obstacles, reaching the goal, then finding the desired target in the Map-3 environment.

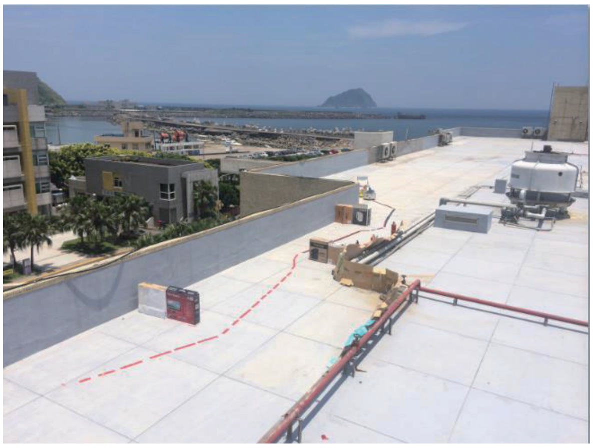


Fig. 18. (Color online) Actual track of the WMR in the Map-3 environment (red dashed line).

shortest path [Fig. 17(h)]. The WMR moves along the shortest path to the third node and maintains a safe distance from the obstacles [Fig. 17(i)], reaches the third turning point, and then turns to the front line to obtain the fourth cluster from the laser sensor [Fig. 17(j)]. The WMR moves in a straight line along the shortest path and maintains a safe distance from the obstacles, reaches the goal point [Fig. 17(k)], and then rotates and searches for the target [Fig. 17(l)]. The WMR finds and approaches the target [Fig. 17(m)], and then reaches the target [Fig. 17(n)]. Figure 18 shows the track of the WMR in the Map-3 environment.

6. Conclusions

We integrated a laser sensor and a DGPS with the A* algorithm and a fuzzy system for collision avoidance and image processing. A laser sensor was utilized to detect the locations of obstacles and provide their edge values. The edge values were applied to the A* algorithm to search for the shortest path for a WMR. Then, the DGPS was used to obtain the WMR and destination positions. A webcam was used to capture the surrounding images, which were transformed from the RGB color space to the HSL color space. The HSL color space was applied to reduce the light interference. An image threshold was set for image binarization, and the median filter was applied to filter out noise in the image. Color segmentation could thus be obtained and the desired color could be captured. The captured color target was then used for tracking control during the rescue task. Obstacle avoidance control was based on a fuzzy system, and the controller was coded using MATLAB. Hardware was implemented using LabVIEW 2010 to realize the human–machine interface. Through the information obtained from the DGPS, the laser measurement, and the webcam, the locations of the WMR and the obstacles were obtained. The distance between the WMR and the desired target was calculated and the WMR was automatically controlled without whole-map information. The desired target was reached by using the positioning, path finding, fuzzy collision avoidance, and image searching. The experiments showed that the proposed control scheme could successfully drive the WMR to a prescribed destination and search for the desired target.

References

1 Y. Zhao and S. L. BeMent: Proc. IEEE Int. Conf. Robotics & Automation (1992) 91–96.

2 Y. P. Leow, K. H. Low, and W. K. Loh: Proc. 7th Int. Conf. Control, Automation, Robotics and Vision (2002) 820–825.

3 Y. Chung, C. Park, and F. Harashima: IEEE Trans. Ind Electron. **48** (2001) 853.

4 W. H. Wang and J. G. Juang: Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (2009) 1677–1682.

5 S. Y. Lin: Master Thesis, SLAM and Navigation in Indoor Environments (National Tsing Hua University, Taiwan, 2011).

6 Y. T. Wang: Master Thesis, Implementation of Robot Visual SLAM Using Low-End Cameras (Tamkang University, Taiwan, 2012).

7 J. L. Martinez, A. Pozo-Ruz, S. Pedraza, and R. Fernandez: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (1998) 204–209.

8 T. Yamakawa, K. Kobayashi, K. Watanabe, and Y. Kurihara: Proc. Annu. Conf. SICE (2010) 892–895.

9 J. L. Chen: Master Thesis, The Study of Robot Environmental Feature Detection in SLAM for a Laser Range Finder (Tamkang University, Taiwan, 2011).

10 A. Srivastava, D. Kartikey, U. Srivastava, and V. Srivastava: Proc. IEEE Conf. Industrial Technology (2010) 553–558.

11 Y. Suzuki, S. Thompson, and S. Kagami: Proc. Int. Conf. Autonomous Robots and Agents (2009) 657–662.

12 E. A. Merchan-Cruz, and A. S. Morris: IEEE Trans. Robotics 22(4) (2006) 613.

13 Y. Zhong, B. Shirinzadeh, and Y. Tian: Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (2008) 1361–1366.

14 E. X. Shi, W. M. Huang, and Y. Z. Ling: Proc. 3rd Int. Conf. Machine Learning and Cybernetics (2004) 439–443.

15 T. H. Lee, H. K. Lam, F. H. F. Leung, and P. K. S. Tam: Proc. 27th Annu. Conf. IEEE Industrial Electronics Society (2001) 570–573.

16 T. H. Lee, H. K. Lam, F. H. F. Leung, and P. K. S. Tam: IEEE Control Systems Magazine **23** (2003) 60.

17 R. Lin, M. Li, and L. Sun: Proc. IEEE Int. Conf. Robotics and Biometics (2013) 1157–1162.

18 Y. H. Kim, B. S. Choi, and J. M. Lee: J. Control, Automation and System Engineering **12** (2006) 106.

19 M. H. A. Hamid, A. H. Adom, N. A. Rahim, and M. H. F. Rahiman: Proc. 5th Int. Conf. Signal Processing & Its Applications (2009) 176–181.

20 S. H. Kim, C. W. Roh, S. C. Kang, and M. Y. Park: Proc. IEEE Int. Conf. Robotics and Automation (2007) 3414–3419.

21 G. Yasuda and B. Ge: Proc. 30th Conf. Industrial Electronics Society (2004) 395–400.

22 A. R. N. Ravari, H. D. Taghirad, and A. H. Tamjidi: Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (2009) 1441–1446.

23 C. Y. Tsai: Ph.D. Thesis, Color Filter Array Interpolation for an Imager Sensor and Visual Tracking Control Design of a Wheeled Mobile Robot (NCTU, Taiwan, 2008).

24 A. Curran and K. J. Kyriakopoulos: Proc. IEEE Mediterranean Symp. New Direction in Control Automation (1993) 8–13.

25 H. S. Chen and J. G. Juang: Proc. National Symp. System Science and Engineering (2008) P0473.

26 Y. H. Lu and J. G. Juang: Sens. Mater. **30** (2018) 1643.

27 M. Ogaz, R. Sandoval, and M. Chacon: Proc. 52nd IEEE Int. Midwest Conf. Circuits and Systems (2009) 306–313.

28 L. Luo and X. Li: Proc. Int. Conf. Image and Graphics (2009) 774–777.

29 R. C. Gonzalez and R. E. Woods: Thresholding. Digital Image Processing (Pearson Education, New York, 2002) 595.

30 A. McAndrew, J. H. Wang, and C. S. Tseng: Introduction to Digital Image Processing with Matlab, Asia Edition, (Cengage Learning Asia Pte Ltd., Taipei, 2010).

31 J. M. Huang: Master Thesis, Vision-Based Auto-Docking System for Mobile Robot (National Yunlin University of Science and Technology, Taiwan, 2010).

32 J. G. Juang and C. W. Lo: Advanced Science Letters **13** (2012) 84.

33 P. Kuang and S. Luo: Proc. IEEE Int. Conf. Wavelet Active Media Technology (2013) 146–148.

34 C. C. Kao and J.G. Juang: Proc. National Symp. System Science and Engineering, (2014) P0223.

35 W. Bai, L. Han, and Y. Shi: Proc. IEEE Int. Conf. Control and Electronic Engineering (2010) 235–237.

36 I. Ullahi, F. Ullah, and Q. Ullah: Proc. Int. Conf. Computer Networks and Information Technology (2011) 241–244.