Journal of Marine Science and Technology

Volume 28 | Issue 5 Article 9

THE RESEARCH OF THE RFID ANCHOR ASSISTED SLAM ON AN AUTONOMOUS PATROLLING SYSTEM

Chien-Wu Lan
Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taoyuan County, Taiwan, R.O.C, chienwulan@gmail.com

Che-Hung Liu
Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taoyuan County, Taiwan, R.O.C.

THE RESEARCH OF THE RFID ANCHOR ASSISTED SLAM ON AN AUTONOMOUS PATROLLING SYSTEM

Acknowledgements

The authors thank the financial support from Ministry of Science and Technology of Taiwan, R.O.C. with Grant number. MOST-108-2221-E-606-015.

Journal of Marine Science and Technology, Vol. 28, No. 5, pp. 394-403 (2020)
DOI: 10.6119/JMST.202010_28(5).0009

THE RESEARCH OF THE RFID ANCHOR ASSISTED SLAM ON AN AUTONOMOUS PATROLLING SYSTEM

Chien-Wu Lan and Che-Hung Liu

Key words: wheeled robot, SLAM, anchor sign-in, autonomous patrol.

ABSTRACT

To allow service robots to patrol indoors is a difficult challenge of the robot control technology. As long as this purpose can be achieved, it can increase the value of service robots in the commercial applications. In this research, an anchor deployment strategy combined with a stereo vision system for the robot synchronization positioning and mapping is proposed, which is based on an open source Gmapping Simultaneous Localization and Mapping (SLAM) algorithm of Robot Operating System (ROS), and enables a robot to simultaneously positioning and build environment maps autonomously. The anchors are developed based on the radio frequency identification (RFID), which can make the robot build the map with a shorter path independently. The proposed system architecture can also reduce the built map error caused by manipulating the robot. Furthermore, the shorter path of the robot moving also reduces the error accumulated by the odometer, and the better quality of the map is obtained. Finally, a built map is combined with the A star path planning algorithm to achieve the autonomous patrolling task. Several experimental results show that maps constructed by anchors have better mapping effects, the robot reaches a target point more accurately, and improve the success rate of sign-in.

I. INTRODUCTION

With the rapid development of technology, the home service robots are increasingly used in the home. The expanding functions based on the existing home machine hardware devices will increase its application value. In Taiwan, the crime population has not improved significantly, and many families hence set up fixed cameras based on security. However, after deployed the fixed cameras, the shortcomings of blind spot will appear. Hence, if the home service robot can perform work on the move while having the autonomous patrol function of security, home security can be further improved.

The issues of moving for home service robots can be divided into three aspects: map construction, positioning and path planning. SLAM is an important technology for the current unknown environmental positioning technology in map construction and location. SLAM enables robots to follow out moving and booking pre-defined tasks in an unknown environment. In the map construction of SLAM, a literature (Asadi and Bozorg, 2009) proposed a decentralized data fusion algorithm that aim at vehicles to simultaneously measure position and establish an environmental map. Li, I. H. et al. propose a monocular based anti-collision path planning method for autonomous robot moving by setting cameras on robot or ceiling (Li et al., 2016). Omara et al. combine Kinect with the open source ROS system to the Gmapping SLAM algorithm to accomplish indoor mapping (Omara and Sahari, 2015). Although the application of ROS-based SLAM algorithm has been widely used, the problem of error in mapping is remain to be modified.

On the other hand, for positioning technology of SLAM, a region segmentation algorithm that distinguishes between foreground (dynamic) and background (static), which is based on the percentage of background points contained in each object to increase the accuracy of positioning (Corcoran et al., 2011). However, if a low-precision sensor is used for positioning, it will cause more uncertainty in the positioning. Therefore, anchor assisted localization has been widely studied. Evers, C. et al. propose a generalized motion (GEM) SLAM method, which is based on a probability hypothesis density filters, and can anchor an observer by fusing motions of the observer and an environment information (Evers and Naylor, 2018). Leitinger, E. et al. use a virtual anchors (VAs) technology to describe a radio reflection situation which is used on the wireless signal SLAM (Leitinger et al., 2019; Yassin et al., 2018). Similar to the previous research, Shi, Q. et al. make use of ultra wideband (UWB) position system with an inertial measurement unit (IMU) to realize a UWB anchor for localization (Shi et al., 2019). In the research of Tinchev, G. et al., positive and negative position samples of 3D information are fused by deep learning to create a representative mark as an

Paper submitted 01/2520; revised 03/25/20; accepted 05/27/20. Corresponding Author: Chien-Wu Lan (e-mail: chienwulan@gmail.com).
Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taoyuan County, Taiwan, R.O.C.

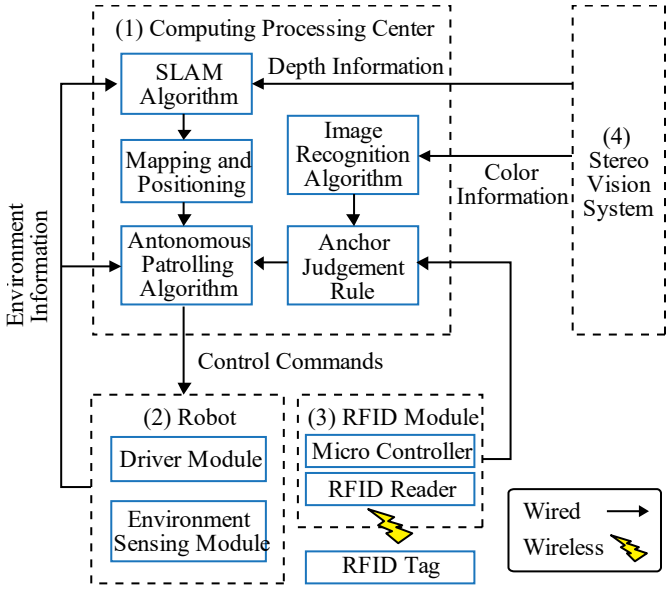

Fig. 1. Architecture of autonomous patrolling system.

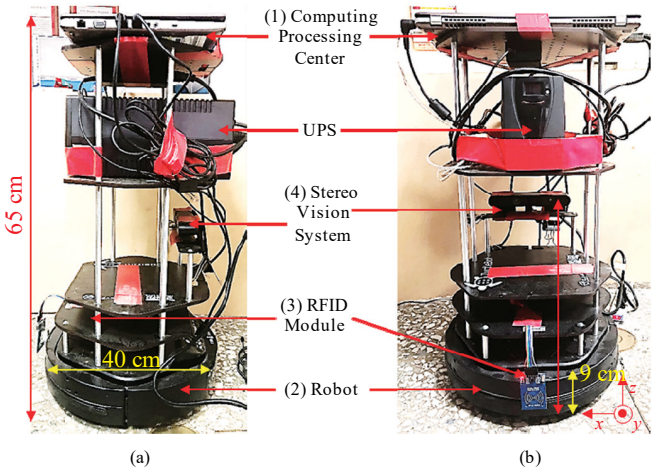

Fig. 2. (a) The side view and (b) the front view of the autonomous patrolling system (Liu, 2018).

anchor for the Laser SLAM (Tinchev et al., 2019). Ventura, J. et al. propose an anchor points in a global map, and use these points to update the local map of client for monocular SLAM (Ventura et al., 2014). In addition, in researches of (Huang et al., 2014) and (Gong et al., 2012), RFID is used as an indoor positioning system to aid in the measurement of the approximate position of an object. Although this method has combined the environmental information of map, it lacks effective assisted orientation technology and cannot improve success rate of sign-in of RFID.

Moreover, in path planning technology of SLAM, a method combined with Hector SLAM, A Star (A*) path planning and dynamic window method makes a robot can automatically reach the specified point in the completed map and avoid obstacles in the path (Chen et al., 2017; Huang, 2016; Shaw et al. 2018). However, there is a lake of mechanism for adjusting the path of robot immediately. Therefore, the problem of odometer accumulation error cannot be solved effectively.

Based on the above researches, the RFID based anchor combines with stereo visual system to assisted SLAM on an autonomous patrolling system is proposed in this paper.

II. THE TRANSCEIVER STRUCTURE

1. System Architecture and Model

The proposed system combined with a computing processing center, a robot, a RFID sensing module end and a stereo vision system (Liu, 2018). The Fig. 1 shows the system architecture. The outline of the architecture is as follows:

(1) Computing processing center:
This computing processing center unit is responsible for the SLAM algorithm, and controls the robot to perform functions such as an autonomous patrol, an anchor judgment and an image recognition via a notebook.

(2) Robot:
By using the Kobuki® home service robot as a vehicle, a patrolling robot combined with a drive module and surrounding sensor modules, and is used to execute the security function of autonomous patrol.

(3) RFID module:
The RFID sensing module fabricated with RFID reader and tag, and is used to implement the function of anchor sign-in.

(4) Stereo vision system:
The Xtion PRO LIVE® camera is used as a stereo visual system for the patrolling robot. The depth information and the color information of the stereo vision system are used for a map construction and an image recognition, respectively.

Fig. 2 shows the integrating hardware of the autonomous patrolling system, and the parts corresponding to the system architecture are marked.

2. Anchors Design and Deployment Strategy

In this paper, an Arduino UNO is used as the micro controller of the RFID sensing module to integrate a MFRC522 RFID Reader, which is used to read the RFID tag of the anchor to realize the patrolling sign-in function. The anchor also contains a red circle with a diameter of 10 cm as the target of the robot's visual system to determine the orientation. On the other hand, the RFID tag deployed on the wall of the patrolling sign-in point is directly below the red circle, and the deployed location is corresponding to the height of the RFID reader of robot to realize the sign-in task. The anchor and patrolling sign-in point is shown in Fig. 3.

After designing the anchor, further, the proposed anchors will be deployed in a particular environment to guide the robot completes the patrol and sign-in tasks in a predefined sequence.

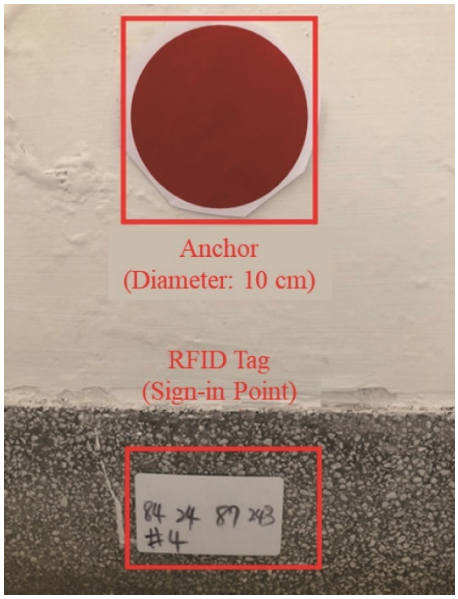

Fig. 3. The design of anchor (Liu, 2018)

This deployment strategy assists the robot to use the stereo vision system and anchor to build the map autonomously, increase the success rate of sign-in, and to solve the error of mapping caused by accumulative error of the odometer.

III. ALGORITHM OF AUTONOMOUS PATROL

1. ROS-Based Gmapping SLAM Algorithm

In this paper, a ROS-based Gmapping SLAM technology (Grisetti et al., 2007; Kamarudin et al., 2015), which is a 2D-SLAM algorithm, is used to construct spatial information around the robot as a mapping application. The Fig. 4 illustrates the ROS-based Gmapping system architecture, which can be divided into three parts: robot hardware, user, and Gmapping SLAM architecture, respectively.

The Gmapping package is originally proposed by Grisettiyz et al. (Grisettiyz et al., 2005), then Hahnel et al. (Hahnel et al., 2003) proposed the Rao-Blackwellized Particle Filter (RBPF), which is using adaptive and resampling techniques to reduce the number of particles required to improve the SLAM. In 2007, it was imported into ROS as an open source SLAM library. The RBPF algorithm uses the Bayesian theorem for factorization. Where, $p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1})$ is the core of Gmapping SLAM algorithm, $x_{1:t}$ is the current attitude of the robot, m is the map, $z_{1:t}$ and $u_{0:t-1}$ are the observation and motion, respectively, $p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1})$ is a particle filter to locate the moving path of the robot, and $p(x_{1:t} \mid z_{1:t}, u_{0:t-1})$ is used to map with a known attitude.

$$p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1}) = p(m \mid z_{1:t}, u_{0:t-1}) \cdot p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \quad (1)$$

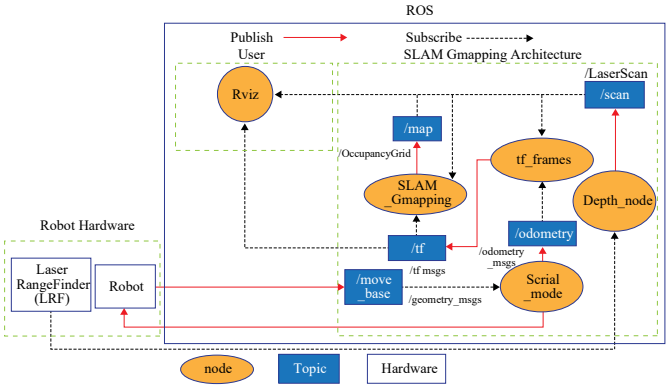

Fig. 4. Architecture of ROS-based Gmapping system (Liu, 2018).

In equation (1), the map built by SLAM has a strong dependence on the position of the robot. The most common particle filter algorithm is the Sampling Importance Resampling (SIR) filter, steps are as follows:

(1) Sampling:

Particles are predicted by equation (2) to produce a large amount of sampled data, which called a particle. Where the variable i is the i-th particle in sample space and x_t^i is the particle set. The algorithm uses the weight values of these particles to weight and approach the Bayesian density.

$$\pi(x_t^i \mid z_{1:t}, u_{0:t}) \quad (2)$$

(2) Correction:

Each particle calculates the corresponding weight value w^i according to formula (3). This value represents the probability that the predicted position acquires particle observation. When the probability of all particles is calculated, the higher the particle weight, the more likely it is to be observed.

$$w^i = \frac{p(x_t^i \mid z_{1:t}, u_{0:t})}{\pi(x_t^i \mid z_{1:t}, u_{0:t})} \quad (3)$$

(3) Resampling:

Particles with lower weights w^i will be replaced and redistributed, and the resampled particles will be input into the next new sampling step to obtain new predicted particles.

(4) Measurement:

For each x_t^i, the corresponding predicted map m_t^i is calculated based on the sampled trajectory and the observed value $p(m_t^i \mid x_{1:t}^i, z_{1:t})$.

While the robot moves, the Gmapping node of ROS subscribes moving distance of an odometer and depth information of stereo vision system from the robot hardware node, and a map around the robot is going to be expand continuously.

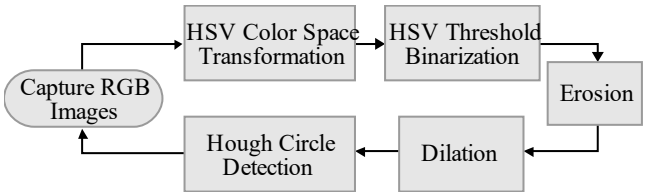

Fig. 5. Image recognition processing flow.

After the map is constructed, the coordinates are converted and positioned to the map by using the distance information, the gyroscope orientation, the wheel encoder information and the vision information provided by the robot to realize the simultaneous positioning and mapping construction. Finally, an environment map required by the patrol system can be obtained by the ROS-based Gmapping algorithm.

Though the ROS-based Gmapping has been applied in many studies, however, there are still shortcomings. For example, the map construction is inefficient and cannot solve the cumulative error of the odometer. Therefore, the proposed method solves this problem by using anchors will be discussed in section three.

2. A* Path Planning Algorithm

The global path algorithm A* (Hart et al., 1968) can obtain the shortest path from the robot to the patrol target point, which is widely used in path planning. The evaluation function of the A* algorithm, as shown in equation (4).

$$f(x) = g(x) + h(x) \tag{4}$$

Where, $f(x)$ is the sum of the distances predicted by any node x, $g(x)$ is the actual distance from the start point to any node x, $h(x)$ represents the predicted distance from any node x to the target point.

First, the predicted distance $h(x)$ of the node x (one of the neighboring nodes of the start point) is estimated by using the start point as the current node after setting a goal in map, and the actual distance $g(x)$ between the start point and the node x is added in $h(x)$ to obtain the sum of the distances $f(x)$.

Secondly, all of neighbor nodes are calculated to obtain the $f(x_{\text{neighbor}})$ dividedly, and find the node with the smallest $f(x_{\min})$ in the neighbor nodes as the node with the shortest path.

Finally, the node $x_{\min}$ is used as the current node to explore all of the neighbor nodes in the next level. According to the method above, the best path of global nodes can be obtained by finding the shortest path recursively until the end point is found. The pseudo code of A* algorithm more detailed procedure is shown as below (Liu, 2018).

Algorithm 1 A* path planning

```
function A* (start, goal)
  closedSet := an empty set
  openSet := the initial node and nodes to be predicted
  cameFrom := an empty map
  gScore[start] := 0
  hScore[start] := heuristic estimate of distance (start, goal)
  fScore[start] := hScore[start]

  while openSet is not empty, do
    currentNode := the node in openSet having the lowest fScore value
    if currentNode = goal then
      return reconstructPath (cameFrom, goal)
    end if
    remove currentNode from openSet
    add currentNode to closedSet
    for each neighbor of currentNode:
      if neighbor in closedSet then
        Continue
        tentative gScore := gScore[currentNode] +
          distant between the currentNode and the neighbor
      end if
      if neighbor not in openSet then
        add neighbor to openSet
        tentative is better := true
      else if tentative gScore < gScore[neighbor] then
        tentative is better := true
        Continue
      else
        tentative is better := false
      end if
      if tentative is better := true then
        cameFrom[neighbor] := currentNode
        gScore[neighbor] := tentative gScore
        hScore[neighbor] := heuristic cost estimate(neighbor, goal)
        fScore[neighbor] := gScore[neighbor] + hScore[neighbor]
      end if
    end for
  end while
end function

function reconstructPath (cameFrom, currentNode)
  totalPath := [currentNode]
  while currentNode in cameFrom, do
    currentNode := cameFrom[currentNode]
    append the currentNode in totalPath
    return totalPath
  end while
end function
```

3. The Flows of Anchor Identification and Sign-in

After the stereo vision system captures the color image, the system will process image according to the flow in the Fig. 5. Further, openCV is used to transform a source image to Hue, Saturation and Value (HSV) color space. The HSV color space can reduce the external interference for the image. Next, the binarization analysis, erosion and expansion of the image are sequentially performed to get images easily to process. Finally, Hough circles detection algorithm is used to detect the circular anchor and solve the center point c_c of the circle, which enables system to identify the anchor in a complex environment and increases the chance of sign-in.

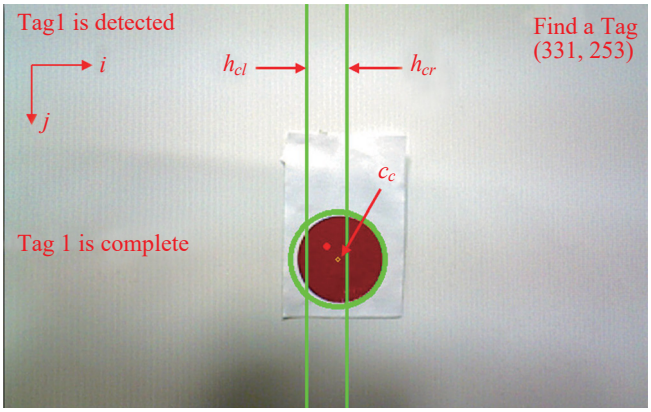

Fig. 6. Anchor identification (Liu, 2018).

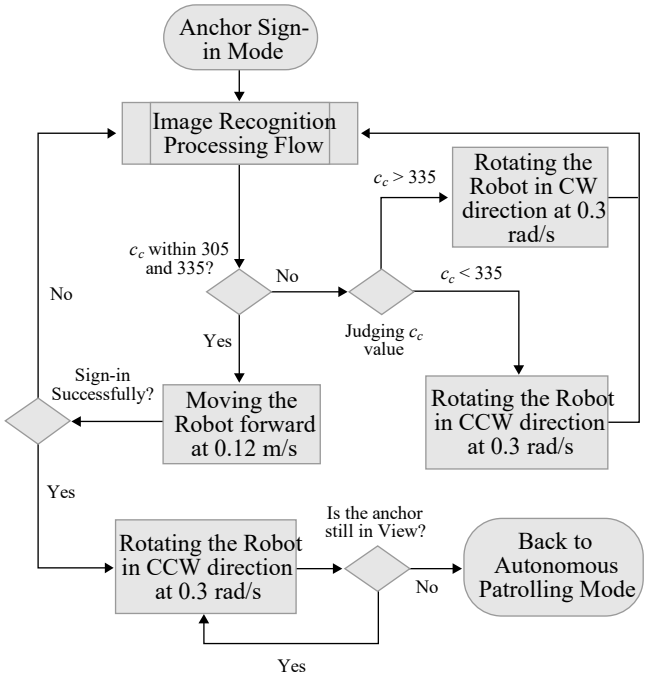

Fig. 7. Flows of sign-in and adjustment.

The Fig. 6 shows an identification results for an anchor. The resolution of the image is 640×480 pixels. Moreover, location between h_{cl} (305) to h_{cr} (335) pixels on x-axis of image is defined as the central area.

After an anchor is detected by the stereo vision system, the robot will further complete the sign-in task according to the flows as following. First, the depth and color images provided by stereo vision system are used to identify the direction of the anchor. Secondly, the robot will move itself to keep the center point c_c of the detected Hough circle to stay in the image's central area. While the center point c_c located in the central area in the image, the robot will move forward. If the center point c_c runs out of the central area on moving, the robot corrects the orientation according to the direction of a unique

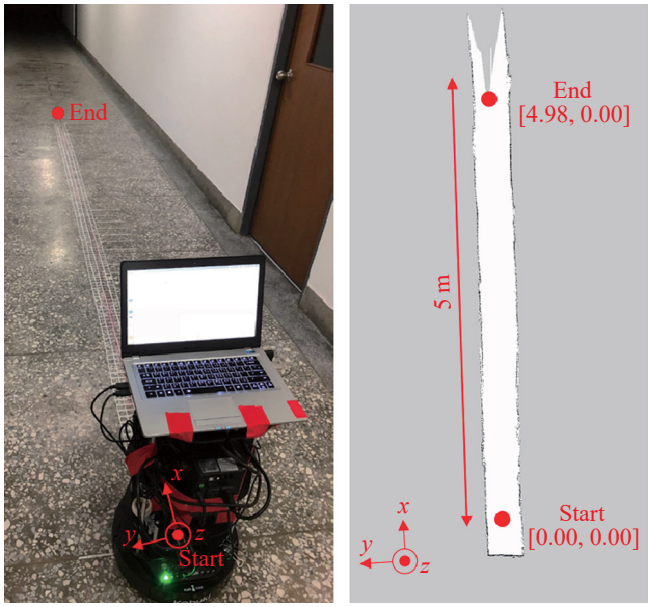

Fig. 8. (a) The actual scene of the simple experiment and (b) the map built by Gmapping function (Liu, 2018).

circular anchor and continues to move forward until the sign-in task is completed. Finally, the robot returns to the autonomous patrolling mode and continue the patrol task after adjusting the robot to complete the sign-in process. The Fig. 7 is showing the flow of the sign-in and adjustment tasks.

To be summarized, in this paper, the concept of semi-automatic map construction (Lee et al., 2015) is used to design the deployment strategy of anchors. The robot completes the sign-in task according to the pre-define anchor respectively. This proposed method can decrease the offset error of odometer caused by the actual long-term movement of the robot and improve the drawback of the displacement of the map.

IV. EXPERIMENT AND RESULTS

To verify the effectiveness of autonomous patrolling system proposed in this paper, two experiments are designed for the mapping results with or without deployment strategy of anchor.

1. Simple Environment

The simple environment experiment is conducted in the straight corridor to understand how the anchor can reduce the path error and the offset distance error from A* path planning, odometer record and real trajectory, and to verify whether the anchors can correctly guide the robot to complete the sign-in and patrol task.

At beginning, an environment map of a straight corridor is built by Gmapping method before the robot moving over it, as shown in Fig. 8. Next, a patrol path from the start point to the end point via the x-axis of the map is planned by the A* algorithm. The direct distance between the start point and the end point is 5 meter with 1 cm resolution, and the coordinate of

Table 1. Comparison of error (without anchor assisted).

Category / Result	Ideal path	Recorded path	Observed path
Stopping point (m)	[4.98, 0.00]	[4.89, 0.06]	[4.86, 0.00]
Length of the moving path (m)	5.04	4.89	4.89
Path length error (%)	N/A	2.98	2.92
Stopping Distance error (m)	N/A	0.11	0.12

Table 2. Comparison of error (with anchor assisted).

Category / Result	Ideal path	Recorded path	Observed path
Stopping point (m)	[4.98, 0.00]	[5.03, 0.01]	[5.01, 0.00]
Length of the moving path (m)	5.04	5.03	5.04
Path length error (%)	N/A	0.15	0.05
Stopping Distance error (m)	N/A	0.05	0.03

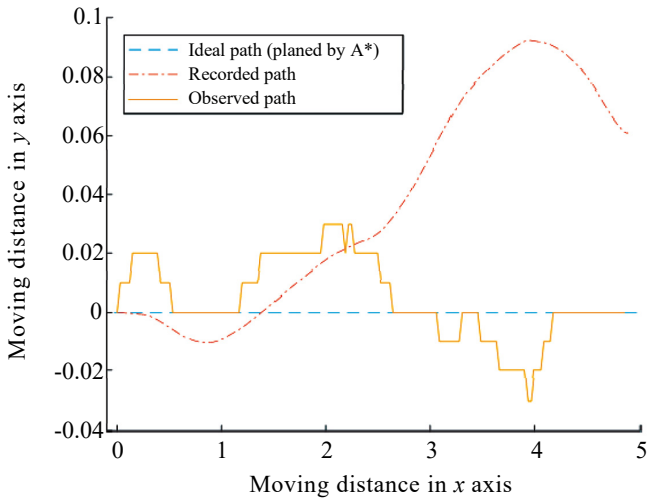

Fig. 9. The result of straight line test.

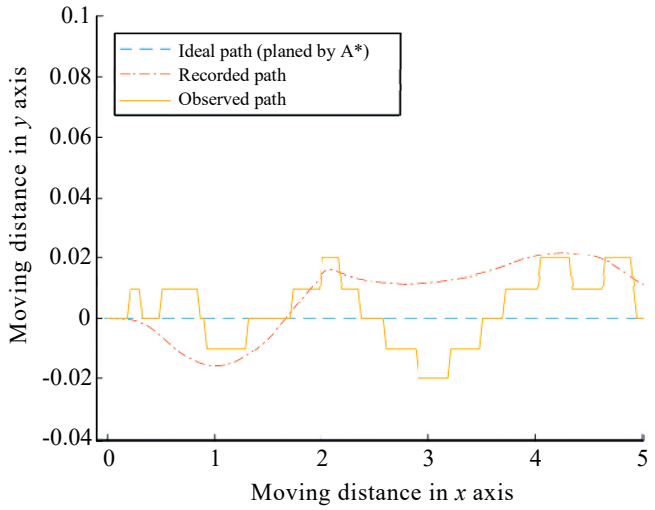

Fig. 10. The result of straight line test guided by anchors.

start point and the end point are defined as [0.00, 0.00] and [5.00, 0.00], respectively. After planning the patrol path, the robot is controlled to move along the designed path. At the meanwhile, the moving distance of the odometer and the actual moving trajectory of robot are recorded. Finally, the designed patrol path by A*, the recorded odometer data and the actual movement are compared with each other.

The Fig. 9 shows the path planned by A*, the recorded odometer path, and the actual moving trajectory of robot in the simple experiment. Though the robot is moving directly toward to the end point, and the trajectory is similar to the A* planned path, the recorded odometer moving path is slightly different from the real trajectory. The situation might be caused by uneven road of the corridor, and the difference between odometer data and the real path will cause an error of SLAM results.

For more, the offset distance from the end point of three path are also compared, too. While the end of the path planned by A* is located on [4.98, 0], the end of the recorded odometer path and the real patrol trajectory are [4.89, 0.06] and [4.86, 0.00], respectively. Hence, the offset distance error from the end point of three path are 2.2% and 0.02%, respectively, as shown in the Table 1. Overall, the error can be controlled below 3%.

On the other hand, in order to realize whether the anchor can improve success rate of sign-in, the simple experiment is re-performed to set the anchor at the end point, and enable the anchor identification and sign-in function of the robot to guide itself to correct the moving path. The Fig. 10 shows the offset distance error of the recorded odometer path and of the real trajectory of robot have been reduced significantly and approach to the path planned by A*. Among them, the improvement of the odometer is the most significant, and the cumulative error of the robot starting from 2.5 meters to end point in the Fig. 9 is greatly reduced. Hence, anchor can greatly reduce the error and suit to autonomous patrolling system.

After the anchor guides robot to the end point, the end position of the recorded odometer and the actual movement are located on [5.03, 0.01] and [5.01, 0.00], respectively. Further, the offset distance error from the end point of odometer and actual movement has been substantially reduced to 0.6% and 0.02% individually, as shown in the Table 2. From the

Table 3. Sign-in experiment in the straight corridor without and with anchor assisted.

Times	Patrolling without anchors assisted		Patrolling with anchors assisted	
	Stopping Point (m)	Sign-in success	Stopping Point (m)	Sign-in success
1	[4.96, 0.03]		[5.03, 0.00]	V
2	[5.03, −0.12]	V	[5.00, 0.00]	V
3	[5.03, 0.03]	V	[4.96, 0.00]	V
4	[4.88, 0.00]		[4.92, 0.05]	V
5	[5.06, 0.03]	V	[5.03, −0.03]	V
6	[4.90, 0.03]		[5.03, 0.03]	V
7	[5.06, 0.00]	V	[4.90, −0.06]	V
8	[4.96, 0.00]		[4.95, −0.03]	V
9	[5.03, 0.00]	V	[5.03, 0.00]	V
10	[4.95, 0.03]		[5.00, 0.00]	V

experimental results, the deployment strategy of the anchor can significantly reduce the offset error of the path and the end point to less than 1%, which can be used to solve the problem of the SLAM cumulative error.

Finally, from the sign-in success rate experiment, this experiment takes anchor existence as the research variable, each performed 10 times sign-in experiments to understand the impact of the anchor existence on the success rate of sign-in. The experiment results are shown in the Table 3 with the coordinate of the stopping points of the path. The Fig. 11 shows that the success rate of completing the task of sign-in is only 50% while the robot getting to the target point through the path plan without anchors assisted. However, with the assistance of anchor allows the robot to achieve a 100% sign-in rate. That is because the robot can determine the direction of the anchor through the vision system during the movement, and correct the path instantly to accurately approach the RFID tag to complete the sign-in task.

2. Simultaneous Positioning and Mapping with Deployment Strategies

After understanding the advantage of the anchors in the simple environment experiment, further, this paper investigate the experiments in the complex environment to understand the similarity between the real map and the map constructed through the deployment strategy. The Fig. 12 shows the expected patrolling path of the robot guided by the deployment strategy with the 1st to 6th anchors to constructing map in the complex experimental environment. During the process of

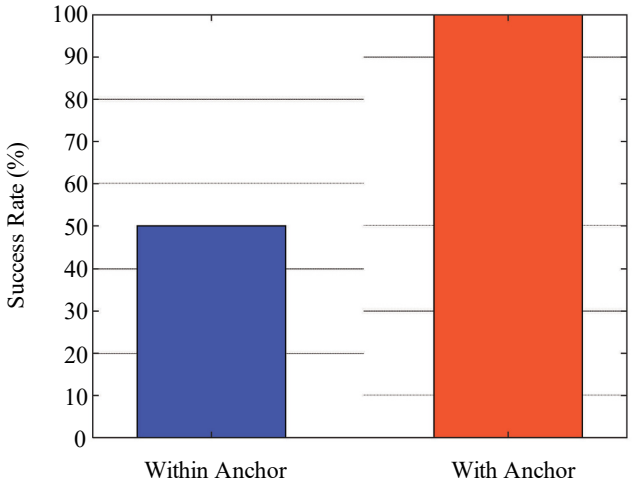

Fig. 11. Success rate of sign-in in the simple environment.

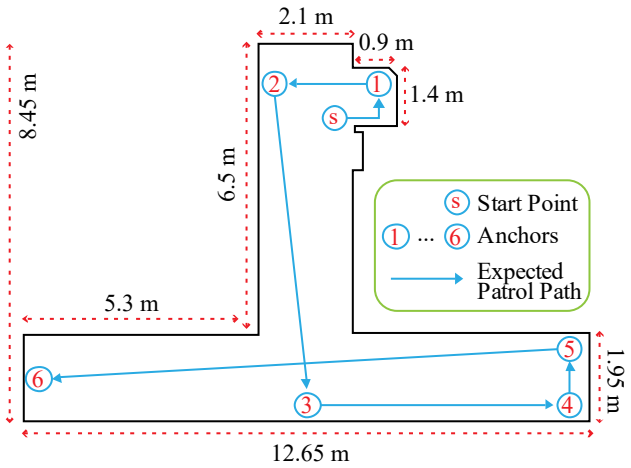

Fig. 12. The complex experimental environment.

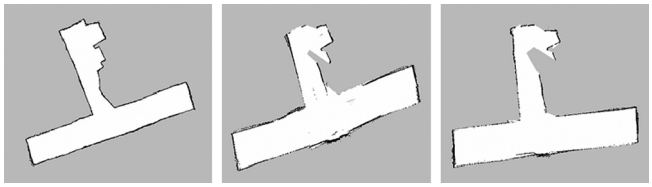

Fig. 13.The maps built (a) manual (as the ground truth) (b) by the obstacle avoidance operation of SLAM, and (c) by the proposed method.

mapping, the Occupancy Grid Map (OGM) is used to present the walkable area and obstacles of the built map.

Since there has no autonomous map building function of Gmapping SLAM, in order to proof the efficiency of the proposed method, the maps of the complex environment built respectively by the proposed method and the autonomous obstacle avoidance operation of Gmapping SLAM are compared in the similarity of the map construction and the completion time.

Table 4. Comparison of Gmapping results and actual maps.

Category	Measured by GIMP (m)	Actual length (m)	Error (m)	Error (%)
The length of each border	1.50	1.40	0.10	7.14%
	0.94	0.90	0.04	4.44%
	2.12	2.10	0.02	0.95%
	6.75	6.50	0.25	3.85%
	2.04	1.95	0.09	4.62%
	12.92	12.65	0.27	2.13%
	2.05	1.95	0.10	5.13%
Average Error (%)				4.04%

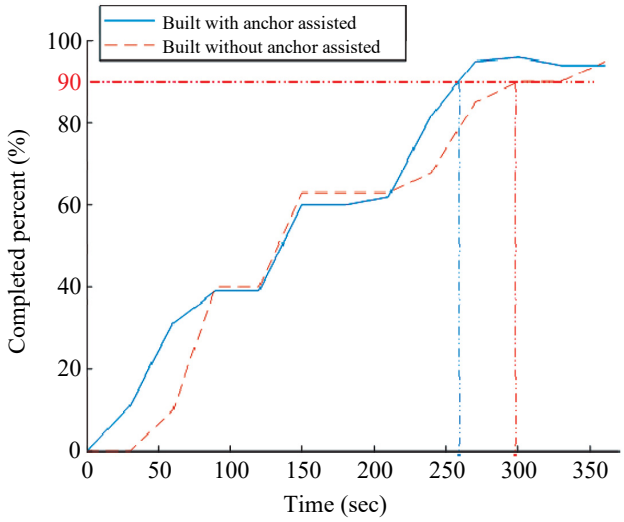

Fig. 14.The completed percent of the maps versus time built with and without anchor assisted, respectively.

At first, the map of the complex environment is built manual as the ground truth, as shown in the Fig. 13 (a). On the other hand, the maps built by the autonomous obstacle avoidance operation function and the proposed method are depicted in the Fig. 13 (b) and (c), respectively. In order to compare the similarity between the built map and the ground truth, the structural similarity (SSIM) method is used (Wang et al., 2004).

The function of similarity SSIM is described in (5), which the $l(\mathbf{X},\mathbf{Y})$ is the luminance, the $c(\mathbf{X},\mathbf{Y})$ is the contrast, and the $s(\mathbf{X},\mathbf{Y})$ is the structure:

$$\mathrm{SSIM}(\mathbf{X},\mathbf{Y}) = l(\mathbf{X},\mathbf{Y}) \cdot c(\mathbf{X},\mathbf{Y}) \cdot s(\mathbf{X},\mathbf{Y}) \tag{5}$$

The functions of luminance $l(\mathbf{X},\mathbf{Y})$, the contrast $c(\mathbf{X},\mathbf{Y})$, and the structure $s(\mathbf{X},\mathbf{Y})$ are also depicted in (6) to (8), which the variable μ_x and μ_y are the pixels average value of two map x and y, the σ_x and the variable σ_y are the pixels standard deviation of two map X and Y, the variable σ_{xy} is the covariance of two map X and Y, the variable r is the range of the pixel value of the maps, which is 255 for the gray scale map, and the coefficient k_1 and k_2 are constants. According to the reference of SSIM, the value of the k_1 and k_2 are set to 0.01 and 0.03, respectively.

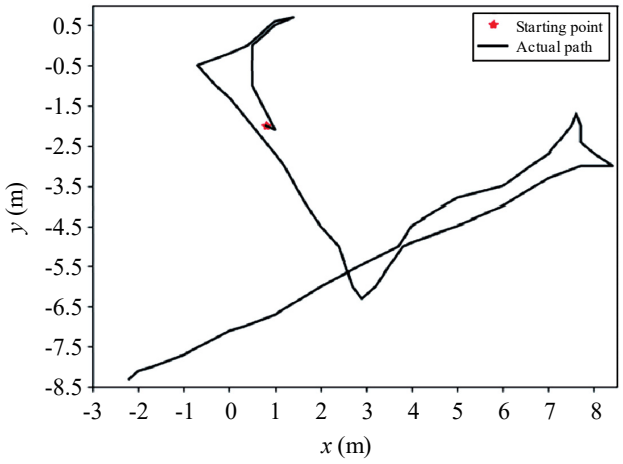

Fig. 15. The actual patrolling path (Liu, 2018).

$$l(\mathbf{X},\mathbf{Y}) = \frac{2\mu_x\mu_y + (k_1 \cdot r)^2}{\mu_x^2 + \mu_y^2 + (k_1 \cdot r)^2} \tag{6}$$

$$c(\mathbf{X},\mathbf{Y}) = \frac{2\sigma_x\sigma_y + (k_2 \cdot r)^2}{\sigma_x^2 + \sigma_y^2 + (k_2 \cdot r)^2} \tag{7}$$

$$s(\mathbf{X},\mathbf{Y}) = \frac{\sigma_{xy} + \frac{(k_2 \cdot r)^2}{2}}{\sigma_x\sigma_y + \frac{(k_2 \cdot r)^2}{2}} \tag{8}$$

Based on the SSIM functions of (6) to (8), the similarity between the map built by the autonomous obstacle avoidance operation (the Fig. 13 (b)) between the ground truth (the Fig. 13 (a)) is 94.835%, and the similarity between the map built by the proposed method (the Fig. 13 (c)) between the ground truth (the Fig. 13 (a)) is 95.167%, respectively. Moreover, the maps completion percentage (or the similarity to the ground truth) versus time are depicted in the Fig. 14, and the 90% completion is used as a benchmark. The percentage of completion of the map built using the recommended method reaches the 90% at about 260 seconds. On the other hand, the time required to reach 90% using the obstacle avoidance method is about 300 seconds.

The actual patrolling path with anchors assisted is depicted in the Fig. 15, which is recorded by the odometer of the robot. The process of map building is depicted in Fig. 16, and the

interval between images is 15 seconds. Moreover, the images shown in the Table 4.

Table 5. Sign-in experiment in the complex environment without and with anchor assisted.

Anchor No.	Patrolling without anchors assisted	Patrolling with anchors assisted
1	Fail	Success
2	Fail	Success
3	Fail	Success
4	Fail	Success
5	Fail	Success
6	Fail	Success

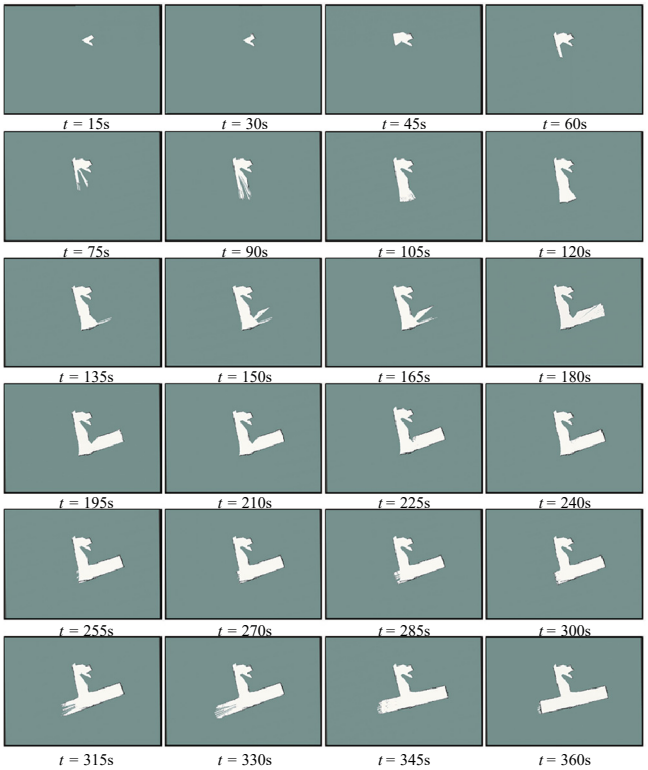

Fig. 16. The map building process with anchor assisted.

on the screen of the computing processing center (left) and the scenes of the robot operating (right) are also depicted in Fig. 17.

In order to compare the accuracy of the map built in this paper, the open source GNU Image Manipulation Program (GIMP) measurement tool is used to measure the distance between pixel points of the image. The distance measured by the GIMP is multiplied by the resolution of the image to be converted into the distance of the map. In this research, the resolution of the built map is 0.03 m per pixel. Finally, comparing the actual border length with the border length of the built map from the Fig. 16, and the accuracy of the comparing result is

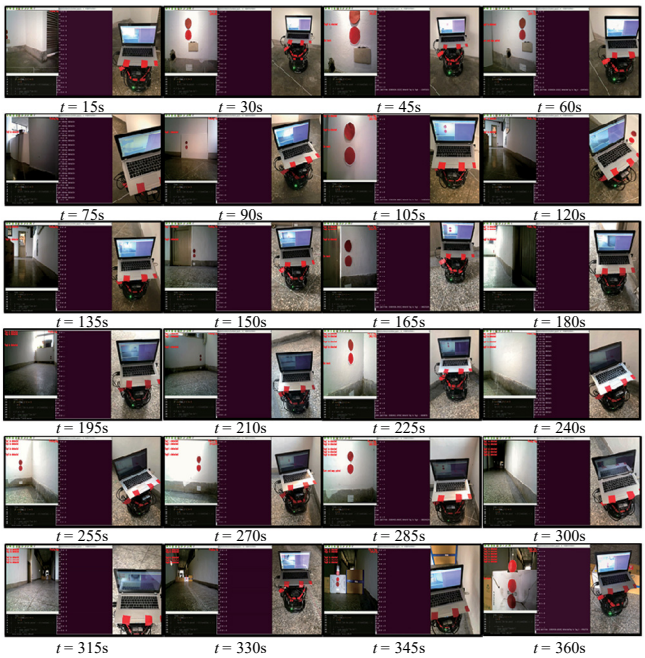

Fig. 17. The screen image (left) and the field image of robot (right) during the map building.

In summary, the map can be completed of a higher proportion and in a shorter time by using the proposed method, and the completed percentage of the built map is higher than 90%. Moreover, the constructed maps with a maximum and minimum error distance of 0.27 meters and 0.02 meters, respectively. Overall, the average error can be controlled to 3.85% for the accuracy of the map constructed by Gmapping SLAM.

Finally, after the map of the complex environment has been built, the success ratio of sign-in at each anchor point is also discussed. For the propose method, all the six anchors are detected successfully during the patrolling. However, all anchors are failed to be detected just by using the navigation function of Gmapping SLAM. The experiment results of the complex environment are depicted in Table 5.

V. CONCLUSIONS

This paper proposes an autonomous patrolling system that extends the functionality of home service robots. With the deployment strategy of anchors and the open source ROS-based Gmapping SLAM technology, which is combined with the stereo vision system and the built-in sensors of the robot to perform in simultaneous positioning and map construction, can realize the autonomous map building and increase the accuracy of sign-in success rate. The proposed system can reduce the cumulative error of the odometer and maintain good accuracy under the autonomous map construction. Finally, the robot to accurately reach the target point and correctly use the RFID sensing module to complete the autonomous

patrolling task.

ACKNOWLEDGMENT

The authors thank the financial support from Ministry of Science and Technology of Taiwan, R.O.C. with Grant number. MOST-108-2221-E-606-015.

REFERENCES

Asadi, E. and M. Bozorg (2009). A decentralized architecture for simultaneous localization and mapping. IEEE/ASME Transactions on Mechatronics 14, 64-71.

Chen, N. Y., J. Shaw and H. I. Lin (2017). Exploration method improvements of autonomous robot for a 2-d environment navigation. Journal of Marine Science and Technology 25, 34-42.

Corcoran, P., A. Winstanley, P. Mooney and R. Middleton (2011). Background foreground segmentation for SLAM. IEEE Transactions on Intelligent Transportation Systems 12, 1177-1183.

Evers, C. and P. A. Naylor (2018). Optimized self-localization for SLAM in dynamic scenes using probability hypothesis density filters. IEEE Transactions on Signal Processing 66, 863-878.

Gong, S., H. Liu, Y. Hu and J. Zhang (2012). ROS-based object localization using RFID and laser scan. 2012 IEEE International Conference on Information and Automation, Shenyang, China, 406-411.

Grisetti, G., C. Stachniss and W. Burgard (2007). Improved techniques for grid mapping with Rao-Blackwellized particle filters. IEEE Transactions on Robotics 23, 34-46.

Grisettiyz, G., C. Stachnissz and W. Burgardz (2005). Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2432-2437.

Hahnel, D., W. Burgard, D. Fox and S. Thrun (2003). An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw Laser range measurements. IEEE Intelligent Robots and Systems, Las Vegas, NV, USA, 206-211.

Hart, P. E., N. J. Nilsson and B. Raphael (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 4, 100-107.

Huang, W. H. (2016). Design of mapping and exploration system with ROS. M.S. thesis, Department of Electrical and Computer Engineering, Tamkang University, Taipei, Taiwan (in Chinese).

Huang, W. Q., D. Chang, S. Y. Wang and J. Xiang (2014). An efficient visualization method of RFID indoor positioning data. The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014), Shanghai, China, 497-504.

Kamarudin, K., S. M. Mamduh, A. S. A. Yeon, R. Visvanathan, A. Y. M. Shakaff, A. Zakaria, L. M. Kamarudin and N. A. Rahim (2015). Improving performance of 2D SLAM methods by complementing Kinect with Laser scanner. 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Langkawi, Malaysia, 278-283.

Lee, C. L., H. T. Chen and Y. B. Lin (2015). Semi-automatically simultaneous localization and mapping for home service robots. 2015 International Conference on Advanced Robotics and Intelligent Systems (ARIS), Taipei, Taiwan, 206-211.

Leitinger, E., F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson and M. Z. Win (2019). A belief propagation algorithm for multipath-based SLAM. IEEE Transactions on Wireless Communications 18, 5613-5629.

Li, I. H., M. C. Chen, W. Y. Wang, S. F. Su and Y. H. Chen (2016). Monocular image-based local collision-free path planning for autonomous robots. Journal of Marine Science and Technology 24, 759-770.

Liu, C. H. (2018). Extend safety requirements and SLAM on home service robots based on RGB-D sensor. M.S. thesis, Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taoyuan, Taiwan (in Chinese).

Omara, H. I. M. A. and K. S. M. Sahari (2015). Indoor mapping using Kinect and ROS. 2015 International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR), Putrajaya, Malaysia, 110-116.

Shaw, J. and W. L. Chi (2018). Automatic classification of moving objects on an unknown speed production line with an eye-in-hand robot manipulator. Journal of Marine Science and Technology 26, 387-396.

Shi, Q., S. Zhao, X. W. Cui, M. Q. Lu and M. D. Jia (2019). Anchor self-localization algorithm based on UWB ranging and inertial measurements. Tsinghua Science and Technology 24, 728-737.

Tinchev, G., A. P. Sanchez and M. Fallon (2019). Learning to see the wood for the trees: deep laser localization in urban and natural environments on a CPU. IEEE Robotics and Automation Letters 4, 1-8.

Ventura, J., C. Arth, G. Reitmayr and D. Schmalstieg (2014). Global localization from monocular SLAM on a mobile phone. IEEE Transactions on Visualization and Computer Graphics 20, 531-539.

Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004), Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13, 600-612.

Yassin, A., Y. Nasser, A. Y. A. Dubai and M. Awad (2018). MOSAIC: simultaneous localization and environment mapping using mmwave without a-priori knowledge. IEEE Access 6, 68932-68947.