ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
A- APPLIED SCIENCES AND ENGINEERING

2019, 20(2), pp. 179 - 194, DOI:10.18038/aubtda.487629

A NOVEL DATA ASSOCIATION TECHNIQUE TO IMPROVE RUN-TIME EFFICIENCY OF SLAM ALGORITHMS

Ziya Uygar YENGİN[1], Volkan SEZER [2, *]

[1] Mechatronics Engineering Department, Graduate School of Science Engineering and Technology, İstanbul Technical University, İstanbul, Turkey
[2] Control and Automation Engineering Department, Faculty of Electrical and Electronics Engineering, İstanbul Technical University, İstanbul, Turkey

ABSTRACT

Simultaneous Localization and Mapping (SLAM) problem is a very popular research area in robotic applications. EKF-SLAM and FastSLAM are widely used algorithms for SLAM problem. The greatest advantage of FastSLAM over EKF-SLAM is that it reduces the quadratic complexity of EKF-SLAM. On the other hand, increasing number of estimated landmarks naturally slows down the operation of FastSLAM. In this paper, we propose a new method called as Intelligent Data Association-SLAM (IDA-SLAM) which reduces this slowing down problem. In data association step also known as likelihood estimation, IDA-SLAM skips comparing a new landmark with all of the pre-calculated landmarks. Instead of this, it compares the newly found one with only nearby landmarks that was found previously. The simulation results indicate that the proposed algorithm significantly speeds up the operation of SLAM without a loss of state estimation accuracy. Real world experiments which have been performed in two different scenarios verify the simulation results. A runtime reduction of 43% and 52% is observed respectively for each of the test environments.

Keywords: Simultaneous localization and mapping, Fastslam, Data association, Particle filter

1. INTRODUCTION

Autonomous robots must overcome various problems such as path planning, obstacle avoidance, human-robot interaction, localization and mapping [1]. Determining the pose while exploring the surrounding without any foreknowledge of the environment and the position is one of the most important tasks for autonomous robots. The SLAM problem also known as concurrent mapping and localization (CML) is a very complicated task because of uncertain sensor readings, slippage of robot legs or wheels and unpredictable environments. In order to cope with these poor conditions, numerous work on SLAM have been done [2].

Many researchers have been interested in SLAM algorithms during last few decades. In 1986, Smith and Cheeseman [3] proposed the basics of the SLAM and they significantly improved their idea after two years [4]. The term simultaneous map building and localization was firstly introduced by Leonard and Durrant-Whyte [5]. Their approach uses Extended Kalman Filter (EKF) for the solution of the SLAM problem. After these studies, at the beginning of millennium, an improved version of [5] called as EKF-SLAM [6] was presented. This new EKF-SLAM method estimates the robot pose and feature locations probabilistically [7]. During two main steps, prediction and correction, EKF-SLAM updates the robot pose and the map. This map consists of a mean vector and a covariance matrix in which all landmarks are correlated with each other. When a new landmark is detected, the covariance matrix grows exponentially and this leads to complexity of $O(M^2)$, where M refers to number of landmarks. Some researchers have been dealt with the scaling problem, however their solutions are effective up to only a few hundred landmarks [8-10]. Another attempt to decrease computational complexity of EKF-SLAM by classifying the landmarks as corners is made by Kim and Park [11]. They identify corners as inward or outward. When a new landmark is detected, it is identified and compared only with ones

*Corresponding Author: sezerv@itu.edu.tr
Receiving: 26.11.2018 Accepted: 12.06.2019

which have same identity. By doing so, the number of landmarks that should be scanned decreases. In the environments such as buildings which consist of numerous corners, their method also slows down. EKF-SLAM is still implemented at various SLAM studies as it is seen in [12] and [13]. Therefore, improving runtime efficiency of EKF-SLAM is very important.

In order to decrease the computational complexity and huge memory requirements in EKF-SLAM algorithms, Rao-Blackwellized particle filter (RBPF) [14] based approaches known as FastSLAM were proposed. This novel technique which was firstly represented by Montemerlo is composed of both particle filter and EKF [15]. The particle filter part is used for estimating the robot pose while the EKF calculates the estimation of landmarks' poses and error ellipses. The main idea of FastSLAM is storing the information of robot pose in a particle vector and also landmarks' means and covariances attached to this vector. In this vector, the landmarks are not associated with each other. It means that when a new landmark is detected, FastSLAM adds a mean and covariance element to the particle vector. Considering this process for all particles in the algorithm, complexity of FastSLAM is bounded by $O(NM)$, where N and M refers to number of particles and landmarks respectively. Even the computational cost and memory requirement in FastSLAM is much lower than EKF-SLAM, the overgrowth of particle set size and number of landmarks are the fundamental disadvantages for its efficiency.

There are lots of approaches aiming to improve the FastSLAM's computational efficiency. Most powerful ones among them were developed by tuning the particle set size, producing effective maps and using landmarks wisely. Kim and Chung brought a different viewpoint to FastSLAM by the method named Unscented FastSLAM (UFastSLAM) that uses nonlinear relations directly without the need of linearization [16]. Their algorithm is more robust and needs less particles than original FastSLAM. Another method for the improvement of particle filter efficiency is changing the number of particles adaptively during the operation [17, 18]. Basically, this idea is based on increasing the number of particles when the uncertainty of posterior distribution is high and decreasing the particles in the contrary case. Studies on computational efficiency are also focused on map management. Designing the full map as sub-maps is another way of speeding up the SLAM process as Chang [19] and Yokozuka [20] were proposed. Additionally, there is an attractive solution for effective map building similar to sub-maps method, using hybrid topological/metric mapping [21]. Handling the map as sub-maps or topologically local maps instead of a full global map brings less memory requirement and computational cost.

Most of the SLAM algorithms face the data association (DA) problem which takes a big portion of run time especially when enormous number of landmarks are found. A novel technique known as CESLAM provides the inspiration for us was proposed by Yang et al [22]. When a new landmark candidate is detected, it needs to be checked if this is really a new one or one of the previous landmarks. The innovation in CESLAM is comparing the new landmark candidate with the landmarks in the range of measurement device, (i.e. a laser range finder (LRF)) instead of all the landmarks in full map at DA step. However an LRF with 360 degree field of view and too long distance range may detect big number of landmarks at any moment. In such a case, the amount of calculations at DA would be high again. Moreover, a landmark that is out of the range of LRF but very close to the newly found one is automatically eliminated from checking process which might cause problems.

In order to solve the mentioned problems, we develop a new approach called as IDA-SLAM, which decides the landmarks to be checked by looking at the landmarks around the newly found one. We take into account the landmarks which are inside a circular area around the newly found one. In this way, we handle less and more efficient landmarks than CESLAM does.

2. PRELIMINARIES OF FASTSLAM

In order to understand the mechanism of FastSLAM clearly, it is important to know the probabilistic basics of SLAM. The robot motion is explained by Markov chain which suggests that the state at time

t can be estimated using the information of the state at time $t-1$ [2]. By looking at this assumption, motion evolution is demonstrated as in equation 1.

$$p(s_t \mid s_{t-1}, u_t) \tag{1}$$

where s_t and u_t indicate the current state and current control input respectively. Furthermore, the robot gets measurements from its environment during the movement. This procedure can also be shown in terms of probabilistic laws as in equation 2.

$$p(z_t \mid s_t, \theta, n_t) \tag{2}$$

where z_t, θ and n_t refer to current measurement, all landmarks estimated from start to current time t and landmark identity at time t respectively. After obtaining the probabilistic functions of motion and measurement, it is straightforward to calculate SLAM posterior which is combination of control inputs and measurements. Using the superscript t to indicate set of variables from start to time t, SLAM posterior is defined as shown in equation 3.

$$p(s^t, \theta \mid z^t, u^t, n^t) \tag{3}$$

While most of the SLAM algorithms such as EKF-SLAM use this posterior directly, FastSLAM factors the equation as illustrated in equation 4.

$$p(s^t, \theta \mid z^t, u^t, n^t) = p(s^t \mid z^t, u^t, n^t) \prod_{n=1}^{N} p(\theta_n \mid s^t, z^t, u^t, n^t) \tag{4}$$

By this means, position of landmarks can be estimated conditionally independent. The most important advantage coming from this factorization is that the FastSLAM updates landmarks' mean and covariance separately for all particles. This makes computational cost very low comparing to EKF-SLAM. Each particle in FastSLAM has the information of robot state and the set of landmarks covariances and means. Hence, the structure of m'th particle is shown in equation 5.

$$S_t^{[m]} = \langle s^{t,[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \ldots, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \rangle \tag{5}$$

where $s^{t,[m]}$ contains all state information of the m'th particle through the operation, $\mu_{k,t}^{[m]}$ is the mean and $\Sigma_{k,t}^{[m]}$ is the covariance of k'th landmark. FastSLAM1.0 defines the new pose of a particle as given in equation 6.

$$s_t^{[m]} \sim p(s_t \mid s_{t-1}^{[m]}, u_t) \tag{6}$$

In some situations such as motion uncertainty is too high relative to the measure ment noise, FastSLAM1.0 may fail at sampling. For this reason, Montemerlo et al. developed a new version of the algorithm called as FastSLAM2.0 which uses both control input and measurement in sampling stage [23]. Thus, the new pose of a particle in FastSLAM2.0 is illustrated in equation 7.

$$s_t^{[m]} \sim p(s_t \mid s^{t-1,[m]}, u^t, z^t, n^t) \tag{7}$$

Another important procedure of FastSLAM algorithm is importance weight estimation. Based on the equations 1 and 4 states of a particle at time t is calculated using $p(s^t \mid z^{t-1}, u^t, n^{t-1})$ which is referred as proposal distribution of particle filtering.This process is done for each particle. The importance weight of m'th particle is estimated by the ratio of the SLAM posterior to proposal distribution as illustrated in equation 8.

$$w_t^{[m]} = \frac{p(s^{t,[m]} \mid z^t, u^t, n^t)}{p(s^{t,[m]} \mid z^{t-1}, u^t, n^{t-1})} \tag{8}$$

As a last step, the particles are resampled with probabilities in proportion to their importance weight values.

3. DATA ASSOCIATION

Basically, matching the newly found features (landmarks) with previous observations is data association. A newly found feature might belong to a new landmark or a previously seen one. This decision provides a fundamental information for particle update process. Since DA is a very critical part of SLAM, lots of studies could be found on the improvement of DA process [24-26].

When a new measurement is obtained, it has to be compared with the expected measurements for all particles. Using the pose of m'th particle and i'th landmark attached to this particle, the expected measurement is estimated as given in equation 9.

$$\hat{z}_i = g(\mu_{i,t-1}^{[m]}, s_t^{[m]}) \tag{9}$$

Same inputs in equation 9 are also used for the calculation of jacobian matrix which is necessary for linearizing as illustrated in equation 10.

$$G_i = g'(\mu_{i,t-1}^{[m]}, s_t^{[m]}) \tag{10}$$

Obtaining the jacobian, it becomes easy to get the innovation covariance matrix as shown in equation 11.

$$Q_i = G_i^T \Sigma_{i,t-1}^{[m]} G_i + R_t \tag{11}$$

where $\Sigma_{i,t-1}^{[m]}$ indicates the covariance of i'th landmark attached to m'th particle and R_t indicates the measurement noise. Finally, the importance weight values can be calculated as given in equation 12.

$$w_i = |2\pi Q_i|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}(z_t - \hat{z}_i)^T Q_i^{-1}(z_t - \hat{z}_i)\right\} \tag{12}$$

After scanning all landmarks of m'th particle, the maximum likelihood is selected to decide the id of landmark. If the likelihood is less than a predetermined threshold value, the measured landmark is assigned as a new landmark.

4. PROPOSED METHOD

In this paper, we develop an improved technique called as IDA-SLAM in order to decrease the run time without loss of the state estimation performance of FastSLAM. In another method CESLAM, the calculation of maximum likelihood of the landmarks are restricted to only ones in range of sensor. The drawbacks of CESLAM are illustrated in coming parts of the paper. By using another strategy than CESLAM, we are interested in decreasing the computational cost of DA process which takes a huge amount of time in case of the number of mapped landmarks are high. Because, the conventional FastSLAM compares all the landmarks with newly detected one which means that the equations in DA step must be performed tons of times.

Idea behind IDA-SLAM algorithm is comparing the landmarks in a circular area whose center is determined by current measurement. As it is seen in Figure 1, when the robot detects a landmark, it creates a circular area with a predefined radius. We call this circular area checking circle along this paper.

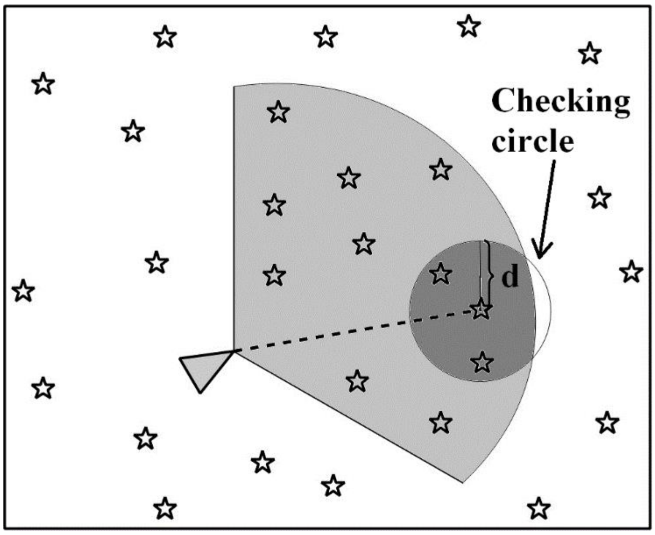


Figure 1. Illustration of proposed method

If a landmark from the map is out of the checking circle, we dont use it for matching with newly detected landmark. Otherwise, we put the new measurement into the DA process. There is only one checking circle for EKF-SLAM and its variants while particle filter based SLAM algorithms include numerous checking circles for each of the particles. In order to estimate the radius of the checking circle, the most uncertain measurement is considered for safety. Because, a newly detected landmarks likelihood depends on the uncertainty of the compared one. Suppose that there is a landmark with fixed distance to the compared landmark. If the uncertainty of the compared landmark is high, the maximum likelihood of the other landmark is high. As the uncertainty of compared landmark decreases, the maximum likelihood also decreases. In order to avoid skipping a necessary landmark in likelihood estimation, we determine the checking circle's radius according to the greatest uncertainty. Because, the uncertainty and the threshold value for likelihood estimation depend on the quality of the measurement device. Deciding the radius of checking circle can be done practically from experiments. The minimum value of the radius should be same as the size of the distance between two landmarks that the sensor can distinguish each other. The overall procedure of proposed method is described as follows:

IDA-SLAM is very similar to FastSLAM1.0. As a first step before taking measurement, the proposal distribution is obtained by using actual control u_t and previous state s_{t-1} for all particles as stated above. After the estimation of proposal distribution for $m = 1 \dots M$, the current measurement is applied to $s_t^{[m]}$ regarding this particles pose and direction. For each particle the estimated measurements are called relative measurement. Figure 2 illustrates how the center of the checking circle is estimated by relative measurement, where the big gray triangle is ground truth of the robots state and smaller triangles are particles. In Figure 2'a, the real measurement is taken by gray triangle (ground truth) and the robot detects a star which symbolizes a landmark. Relative measurement is estimated for the third particle and the circle is determined with radius d. In this situation, there are three suitable landmarks to put into the DA process among eight landmarks in the range of measurement of the first particle. In Figure 2'b, real measurement with the angle ϕ and distance r is adapted to one of the particles.

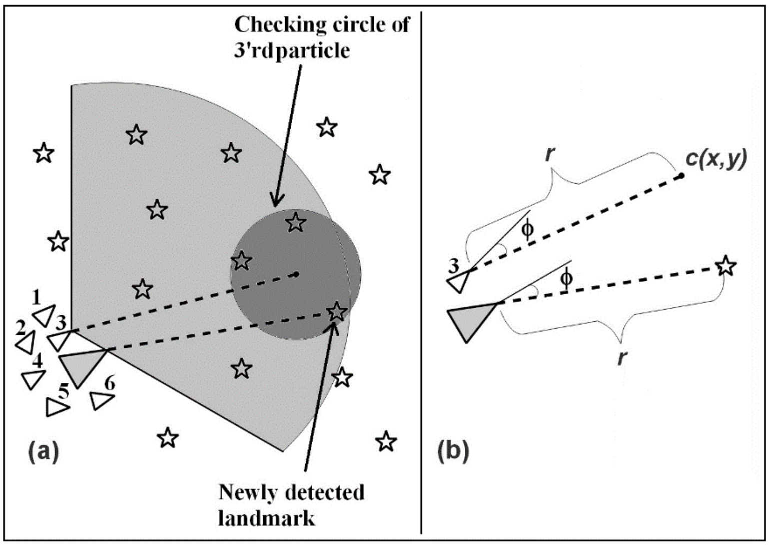


Figure 2. Estimating center of checking circle

Center of the checking circle in $x - y$ plane is obtained by a simple geometric relation as shown in equations 13 and 14.

$$c_x = s_{t,x}^{[m]} + r\,cos(s_{t,theta}^{[m]} + \phi) \tag{13}$$

$$c_y = s_{t,y}^{[m]} + r\,sin(s_{t,theta}^{[m]} + \phi) \tag{14}$$

where c_x is x coordinate, c_y is y coordinate of the checking circles center and $s_{t,theta}^{[m]}$ is heading angle of the particle. After calculating the center for m 'th particle, all landmarks of this particle are scanned if they are in the checking circle or not. For $n = 1 \dots N$, the distance l_n between l'th landmark and the center of circlecan be calculated as illustrated in equation 15.

$$l_n = \sqrt{(c_x - \mu_{\mathrm{n,x}}^{[m]})^2 + (c_y - \mu_{\mathrm{n,y}}^{[m]})^2} \tag{15}$$

If l_n is greater than the radius of center d, the landmark is skipped. and the newly measured landmark is assigned as a new one for the m'th particle. Otherwise, it is evaluated in the standard DA process. After all landmarks are checked if they are in the checking circle and the DA process is implemented for ones in the checking circle, the most probably landmark is assigned to measured landmark. If non of the landmarks are in the checking circle, the measured landmark is assumed as a new one. When this procedure is performed for all particles in order, the remaining part of the algorithm is same as the FastSLAM1.0. If the measured landmark is a new landmark, it is attached to the m’th particle with its covariance and mean; if it is a known landmark, the mean and covariance values are updated. Finally, all the weighted particles processed in DA step are resampled in proportion to their weights. The complete mechanism of proposed method can clearly be understood from the algorithm illustrated in Table 1.

Table 1. IDA-SLAM

```
for all particles
    sample new pose
    get measurement
    for all estimated landmarks
        estimate $(c_x, c_y)$
        estimate $l_n$
            if $l_n < d$
                estimate maximum likelihood $(p_n)$
            else $p_n = 0$
            end
    end
            if $p_n >$ threshold
                update landmark
            else
                assign as a new landmark
            end
    estimate importance weight
end
resample
```

Even the strategy seems similar in CESLAM, the CESLAM may miss some important landmarks to be checked or relatively unimportant landmarks could be checked unnecessarily. Figure 3 shows such kind of scenario. As it is seen in Figure 3, the robot has a new landmark measurement that was also found previously (landmark-1). When the measurement is adapted to the particle (small triangle), CESLAM ignores the previously found landmark-1 for checking process since it is just out of the measurement range. This is one of the drawbacks of CESLAM approach and effects the accuracy of the method badly. Another problem of CESLAM is it checks landmark-3 which is far away from the measurement and not very necessary. On the other hand IDA-SLAM draws a checking circle around the landmark candidate which prevents these problems. As it is seen in Figure 3, the particle takes into account both of the landmark-1 and landmark-2 but not landmark-3 for checking process.

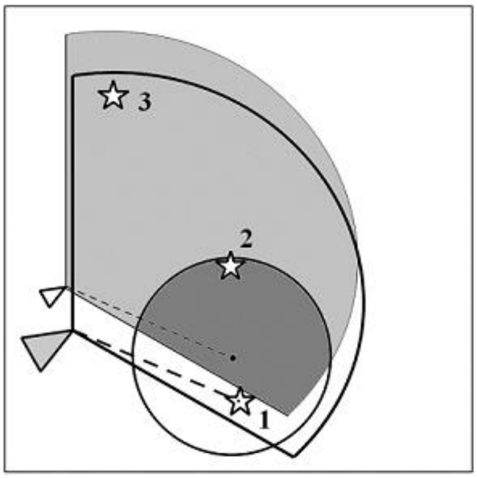


Figure 3. Comparison of CESLAM and IDA-SLAM for a specific scenario

5. SIMULATION AND EXPERIMENTAL RESULTS

After the development and coding process, the proposed method is tested both in simulation and real experimental tests. The runtime values, robot trajectory and landmarks pose estimations are all evaluated during these tests. We compare IDA-SLAM with FastSLAM1.0 and CESLAM in order to understand the effect of checking circle clearly. The simulations are carried out using a differential drive robot platform Turtlebot with a Kinect sensor shown in Figure 4. Kinect is an RGB-D Sensor and provides a point cloud data which is widely used in computer vision applications [27].

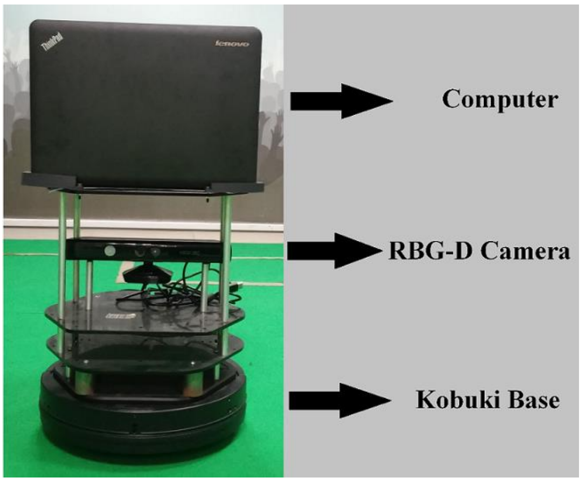


Figure 4. Turtlebot 2 Robotic Test Platform

In order to create realistic simulation environments, Gazebo simulator [28] is used under Robot Operating System (ROS). Additionally, Open Source Computer Vision (OpenCV) library [29] provides us visualizing the landmarks and the robot trajectory. In Gazebo simulator, two different environments are created using cube shaped objects as shown in Figures 5 and 6. The first environment has 16 objects where the Turtlebot detects 16 landmarks while operating; while the second one has 20 objects where the Turtlebot detects 23 landmarks. The reason for the selection of cubic objects is that they can be easily distinguished by Kinect as landmarks by using our landmark detection technique. We use the curvature function based method in [30] which makes it easy to identify the geometric shapes using two-dimensional laser data. Generally; corners, walls and circular objects can be recognised by means of this technique. Turtlebot moves with 500 particles in both of the environments and follows a square route whose side length is 8 meters. The robot goes around this square route two times at each operation. All numerical results indicate the average values obtained from 30 independent runs.

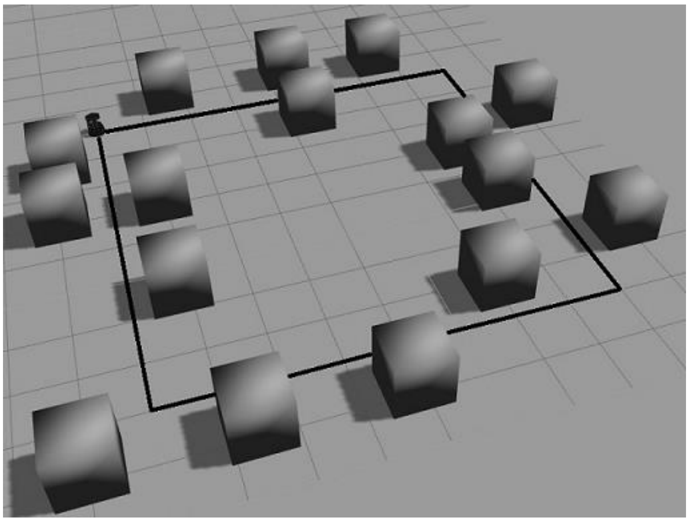


Figure 5. Simulation environment-1 created in Gazebo

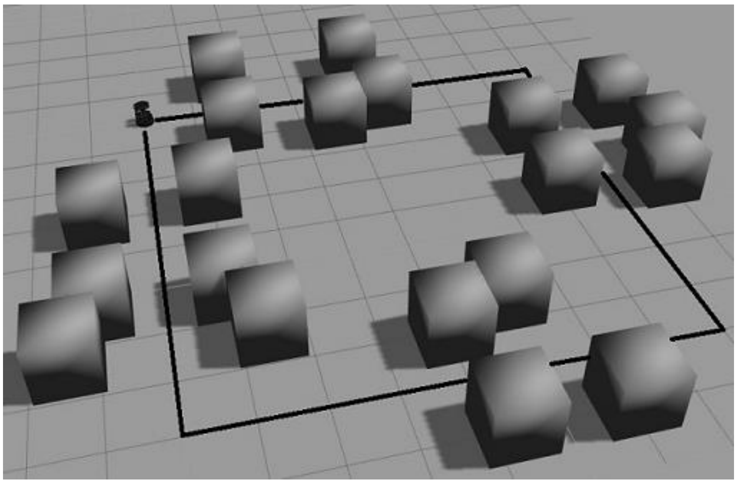


Figure 6. Simulation environment-2 created in Gazebo

Figures 7 and 8 show the simulation results of IDA-SLAM tested in two different environments, where yellow points are estimated position of landmarks and ellipses represent the uncertainty of estimated landmarks. Additionally, black lines are estimated paths of the robot.

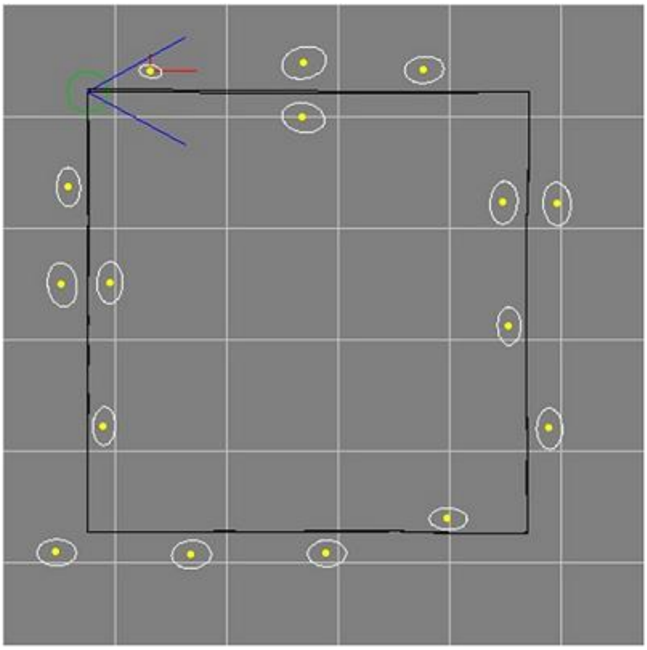


Figure 7. Simulation result of IDA-SLAM with 16 landmarks

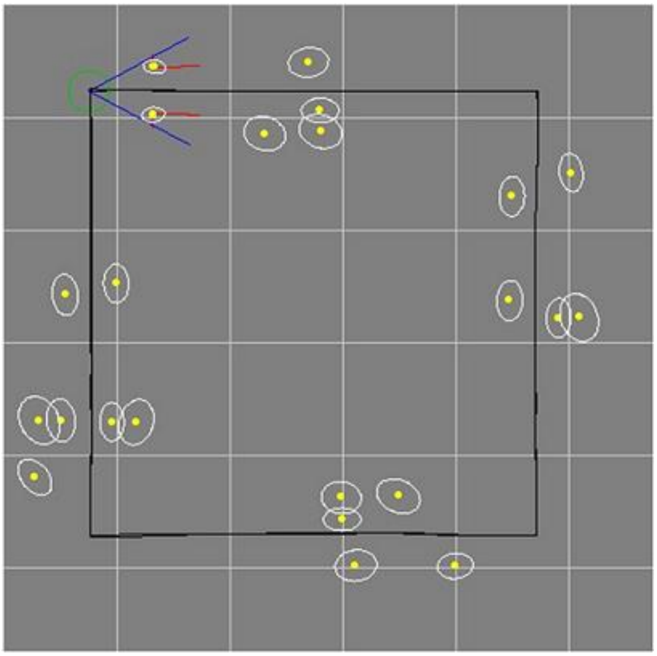


Figure 8. Simulation result of IDA-SLAM with 23 landmarks

Table 2 indicates the average error values of estimated landmarks for each three SLAM algorithms implemented in two different simulation environments. The landmark error values (e_{land}) are calculated regarding the differences between the ground truth and estimated locations of landmarks as it shown in equation 16, where μ_{re}^{i} is real pose of i'th landmark and μ_{re}^{i} is estimated pose of i'th landmark. As it is expected, there are no distinct differences between FastSLAM1.0, CESLAM and IDA-SLAM in terms of error values.

Table 2. Average landmark pose errors of various SLAM algorithms

	Landmark Error (Environment 1)			**Landmark Error (Environment 2)**		
SLAM Algorithms	**Mean (m)**	**Standart Deviation**	**Maximum Value (m)**	**Mean (m)**	**Standard Deviation**	**Maximum Value (m)**
FastSLAM1.0	0.068	0.01	0.096	0.098	0.006	0.128
CESLAM	0.070	0.044	0.082	0.095	0.004	0.103
IDA-SLAM	0.068	0.045	0.079	0.094	0.005	0.104

Similar to the landmark estimation, the performance of path estimation of each algorithms are also similar as it is illustrated in Table 3.

Table 3. Average path errors of various SLAM algorithms

	Path Error - Simulation Env.1			**Path Error - Simulation Env. 2**		
SLAM Algorithms	**Mean (m)**	**Standart Deviation**	**Maximum Value (m)**	**Mean (m)**	**Standard Deviation**	**Maximum Value (m)**
FastSLAM1.0	9.64	3.25	18.65	10.79	4.57	18.63
CESLAM	9.73	2.70	17.76	10.56	2.71	15.98
IDA-SLAM	9.12	3.31	17.09	9.71	3.10	15.56

Path errors are calculated by accumulating the pose errors of particles at each time step from beginning to end of an individual run. Mathematical illustration of the path error estimation (e_{path}) is shown in equation 17, where s_{re}^{t} is real pose of robot and s_{est}^{t} is estimated pose of robot.

$$e_{land} = \sum_{i=1}^{N} \sqrt{(\mu_{re,x}^{i} - \mu_{est,x}^{i})^2 + (\mu_{re,y}^{i} - \mu_{est,y}^{i})^2} \tag{16}$$

$$e_{path} = \sum_{t=0}^{T} \sqrt{(s_{re,x}^{t} - s_{est,x}^{t})^2 + (s_{re,y}^{t} - s_{est,y}^{t})^2} \tag{17}$$

Finally, Table 4 compares the runtime efficiencies of each method which is the main focus of this paper. Since the standard FastSLAM method checks all the landmarks when a new measurement is obtained, it is more awkward than CESLAM and IDA-SLAM. This result can be seen from Table 4 that CESLAM and IDA-SLAM operate more than 2 times faster than FastSLAM.

Table 4. Average runtime values of various SLAM algorithms

	Runtimes - Simulation Env.1			**Runtimes - Simulation Env. 2**		
SLAM Algorithms	**Mean (s)**	**Standart Deviation**	**Maximum Value (s)**	**Mean (s)**	**Standard Deviation**	**Maximum Value (s)**
FastSLAM1.0	13.88	1.09	16.24	27.33	1.29	30.93
CESLAM	4.88	0.33	5.83	8.11	0.54	9.15
IDA-SLAM	4.73	0.33	5.63	6.16	0.38	7.21

When we compare our algorithm with CESLAM, it seems that IDA-SLAM is more efficient when the map becomes complicated. According to the results, IDA-SLAM is 66% and CESLAM is 65% faster than FastSLAM1.0 when there are 16 landmarks in the map. Furthermore, IDA-SLAM is 77% and CESLAM is 70% faster than FastSLAM1.0 when there are 23 landmarks in the map. Checking circle approach of IDA-SLAM makes it more efficient comparing to the other methods.

In Figures 9, 10 and 11, it can be seen the distributions of landmark estimation errors, path errors and runtimes for different SLAM methods and environments.

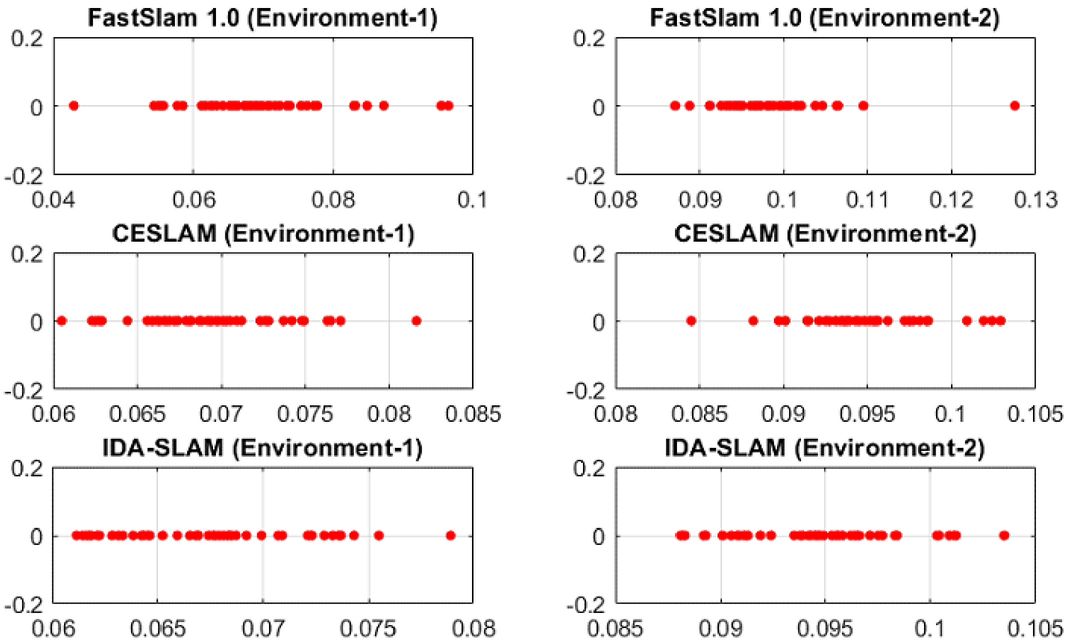


Figure 9. Landmark estimation error distributions for environment-1 (left) and environment-2 (right)

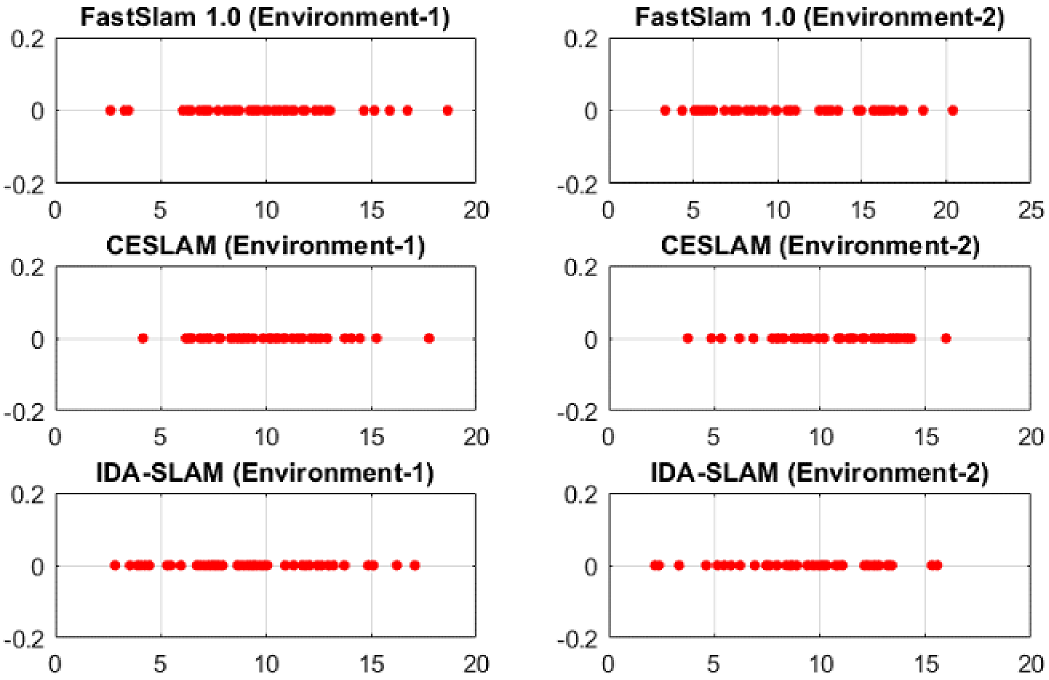


Figure 10. Path error distribution for environment-1 (left) and environment-2 (right)

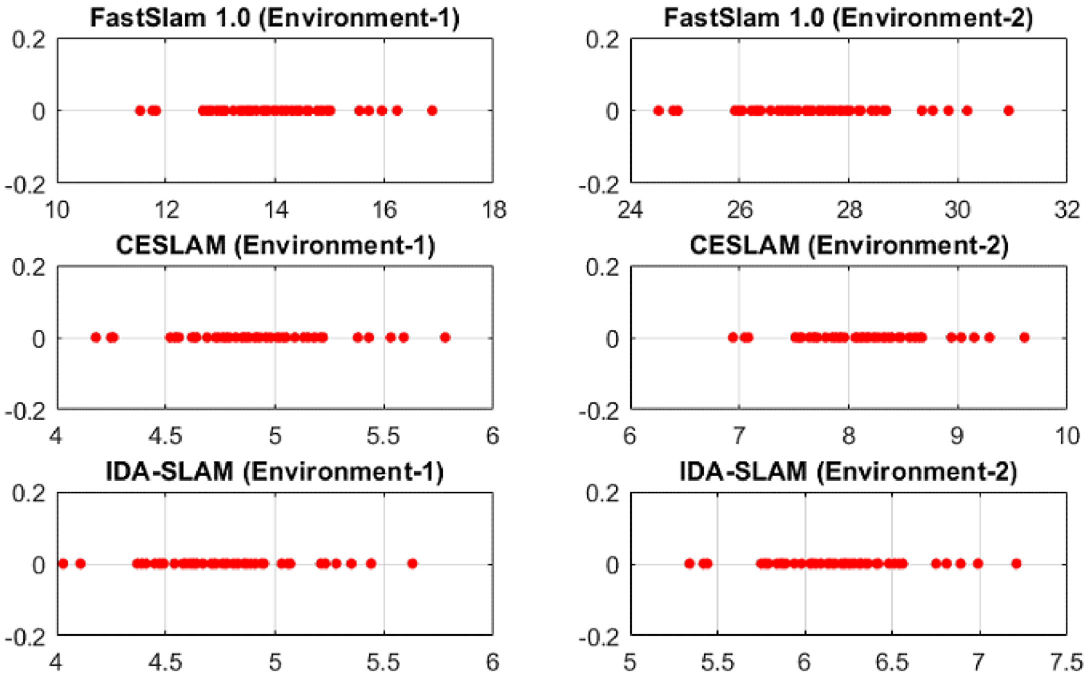
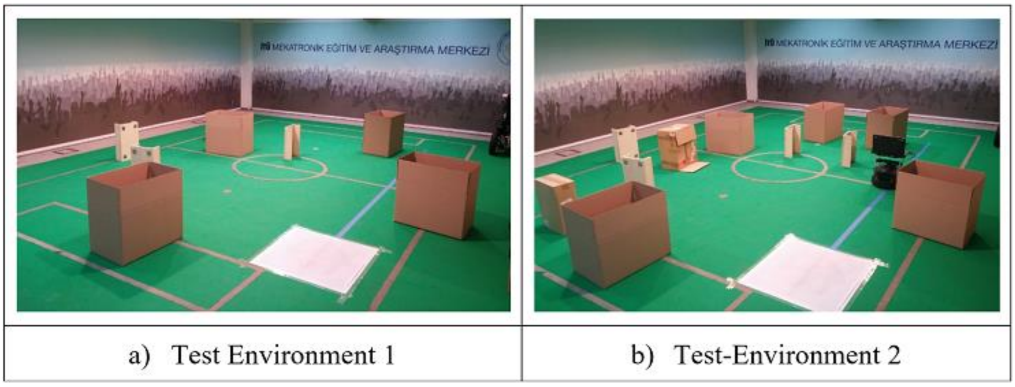


Figure 11. Runtime distribution of IDA-SLAM for environment-1 (left) and environment-2 (right)

As shown in Figure 12, the real environments which are arranged in a laboratory at Istanbul Technical University are similar to ones created in Gazebo. The robot moves with 100 particles in both of the environments and follows a square route whose side length is 3 meters. The robot goes around this square route two times at each operation same as the simulations.

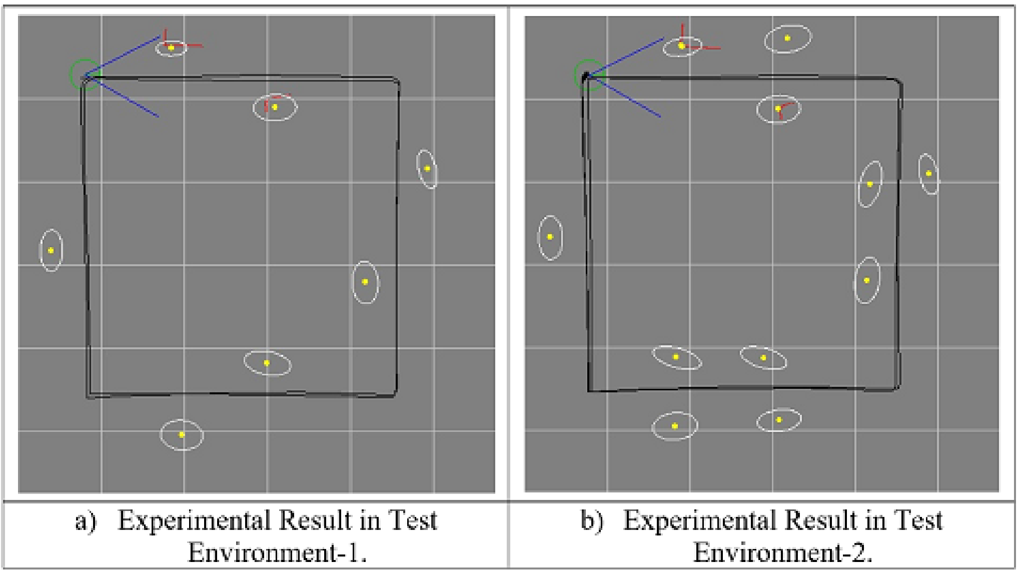

MEKATRONİK EĞİTİM VE ARAŞTIRMA MERKEZİ
MEKATRONİK EĞİTİM VE ARAŞTIRMA MERKEZİ

a) Test Environment 1

b) Test-Environment 2

Figure 12. Real world test environment

In Figure 13, two different maps created by IDA-SLAM in test environment-1 and test environment-2 are shown. The black line indicates the estimated poses at the operation and yellow dots indicate the estimated landmarks' poses.



a) Experimental Result in Test Environment-1.

b) Experimental Result in Test Environment-2.

Figure 13. Experimental result of IDA-SLAM in real world environments

Table 5 shows the runtime results and improvement rates of different SLAM algorithms performed in test environment-1 and test environment-2. According to the results, IDA-SLAM is 43% and CESLAM is 36% faster than FastSLAM1.0 when there are 7 landmarks in the map. Also, IDA-SLAM is 52% and CESLAM is 43% faster than FastSLAM1.0 when there are 11 landmarks in the map. It is clearly understood from the test results that the proposed method improves the runtime efficiency comparing to the other methods. The difference of the improvement rates comes from the structure of the maps. In more complicated environments, number of landmarks that the robot detect at a single measurement are more than the simpler environments. For this reason, the proposed method becomes more effective in such kind of crowded environments.

Table 5. Runtime values of various SLAM algorithms in real experimental tests

SLAM Algorithms	**Test Environment-1**		**Test Environment-2**	
	Runtime(sec)	**Improvement Rate**	**Runtime(sec)**	**Improvement Rate**
FastSLAM1.0	0.81		1.22	
CESLAM	0.52	36%	0.69	43%
IDA-SLAM	0.46	43%	0.58	52%

6. CONCLUSION

This paper proposes the IDA-SLAM algorithm which brings a new solution to the landmark matching problem. In DA step, IDA-SLAM doesn't consider the landmarks which are too far from newly measured landmark. Creating a virtual circle around the landmark candidate, restricts the area to be scanned. By this way, IDA-SLAM avoids calculating likelihood and other matrices for each of the landmarks in DA process. Simulation results show that the proposed approach improves the runtime efficiency while the performance of state estimation remains almost same. The real experimental tests also verify the runtime efficiency of IDA-SLAM.

The new approach brings a new solution to DA process which is common in all SLAM algorithms. For this reason, the new solution can be implemented on not only particle based methods, but also EKF based SLAM algorithms. In addition to FastSLAM, the proposed method can be performed on EKF-SLAM and the comparison of the results with conventional techniques can be analyzed as a future work.

ACKNOWLEDGEMENTS

This work was supported by Research Fund of the Istanbul Technical University. Project Number: 41399.

REFERENCES

[1] Tzafestas SG. Mobile Robot Control and Navigation: A Global Overview. Journal of Intelligent & Robotic Systems. 2018:1-24.

[2] Burgard W, Fox D, Thrun S. Probabilistic robotics. The MIT Press. 2005.

[3] Smith RC, Cheeseman P. On the representation and estimation of spatial uncertainty. The international journal of Robotics Research. 1986 Dec; 5(4):56-68.

[4] Smith R, Self M, Cheeseman P. Estimating uncertain spatial relationships in robotics. InAutonomous robot vehicles 1990 (pp. 167-193). Springer, New York, NY.

[5] Leonard JJ, Durrant-Whyte HF. Simultaneous map building and localization for an autonomous mobile robot. In: Proc. IEEE Int. Workshop on Intelligent Robots and Systems (IROS); 3-5 November 1991; Osaka, Japan. pp. 1442-1447.

[6] Dissanayake G, Newman P, Clark S, Durrant-Whyte HF and Csorba M. An experimental and theoretical investigation into simultaneous localisation and map building (SLAM). Lecture Notes in Control and Information Sciences: Experimental Robotics VI, Springer, 2000.

[7] Wen S, Sheng M, Ma C, Li Z, Lam HK, Zhao Y, Ma J. Camera Recognition and Laser Detection based on EKF-SLAM in the Autonomous Navigation of Humanoid Robot. Journal of Intelligent & Robotic Systems. 2017:1-3.

[8] Guivant JE, Nebot EM. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. IEEE transactions on robotics and automation. 2001 Jun;17(3):242-57.

[9] Leonard JJ, Feder HJ. A computationally efficient method for large-scale concurrent mapping and localization. InRobotics Research 2000 (pp. 169-176). Springer, London.

[10] Lu F, Milios E. Globally consistent range scan alignment for environment mapping. Autonomous robots. 1997 Oct 1;4(4):333-49.

[11] Kim TH, Park TH. EKF-based simultaneous localization and mapping using laser corner-pattern matching. InInformation and Automation (ICIA), 2016 IEEE International Conference on 2016 Aug 1 (pp. 491-497). IEEE.

[12] Saman AB, Lotfy AH. An implementation of SLAM with extended Kalman filter. InIntelligent and Advanced Systems (ICIAS), 2016 6th International Conference on 2016 Aug 15 (pp. 1-4). IEEE.

[13] Deng G, Li J, Li W, Wang H. SLAM: Depth image information for mapping and inertial navigation system for localization. InIntelligent Robot Systems (ACIRS), Asia-Pacific Conference on 2016 Jul 20 (pp. 187-191). IEEE.

[14] Murphy KP. Bayesian map learning in dynamic environments. InAdvances in Neural Information Processing Systems 2000 (pp. 1015-1021).

[15] Montemerlo M, Thrun S, Koller D, Wegbreit B. FastSLAM: A factored solution to the simultaneous localization and mapping problem. Aaai/iaai. 2002 Jul 28;593598.

[16] Kim C, Sakthivel R, Chung WK. Unscented FastSLAM: a robust and efficient solution to the SLAM problem. IEEE Transactions on Robotics. 2008 Aug; 24(4):808-20.

[17] Kwak N, Kim IK, Lee HC, Lee BH. Adaptive prior boosting technique for the efficient sample size in FastSLAM. InIntelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on 2007 Oct 29 (pp. 630-635). IEEE.

[18] Xu W, Jiang R, Xie L, Tian X, Chen Y. Adaptive square-root transformed unscented FastSLAM with KLD-resampling. International Journal of Systems Science. 2017 Apr 26; 48(6):1322-30.

[19] Chang HJ, Lee CG, Lu YH, Hu YC. A computational efficient SLAM algorithm based on logarithmic-map partitioning. InIntelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on 2004 Sep (Vol. 2, pp. 1041-1046). IEEE.

[20] Yokozuka M, Matsumoto O. Sub-map dividing and re-alignment FastSLAM with scalable voxel map system. InAdvanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on 2012 Jul 11 (pp. 180-185). IEEE.

[21] Kouzoubov K, Austin D. Hybrid topological/metric approach to SLAM. InRobotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on 2004 Apr (Vol. 1, pp. 872-877). IEEE.

[22] Yang CK, Hsu CC, Wang YT. Computationally efficient algorithm for simultaneous localization and mapping (SLAM). InNetworking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on 2013 Apr 10 (pp. 328-332). IEEE.

[23] Stentz A, Fox D, Montemerlo M. Fastslam: A factored solution to the simultaneous localization and mapping problem with unknown data association. InIn Proceedings of the AAAI National Conference on Artificial Intelligence 2003.

[24] Hähnel D, Thrun S, Wegbreit B, Burgard W. Towards lazy data association in SLAM. InRobotics Research. The Eleventh International Symposium 2005 (pp. 421-431). Springer, Berlin, Heidelberg.

[25] Montemerlo M, Thrun S. Simultaneous localization and mapping with unknown data association using FastSLAM. InICRA 2003 Sep 14 (pp. 1985-1991).

[26] Weingarten J, Siegwart R. EKF-based 3D SLAM for structured environment reconstruction. InIntelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on 2005 Aug 2 (pp. 3834-3839). IEEE.

[27] Han J, Shao L, Xu D, Shotton J. Enhanced computer vision with microsoft kinect sensor: A review. IEEE transactions on cybernetics. 2013 Oct; 43(5):1318-34.

[28] Koenig NP, Howard A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. InIROS 2004 Sep 28 (Vol. 4, pp. 2149-2154).

[28] Bradski G. OpenCV: Examples of use and new applications in stereo, recognition and tracking. InProc. Intern. Conf. on Vision Interface (VI'2002) 2002 May (p. 347).

[30] Núñez P, Vázquez-Martín R, Del Toro JC, Bandera A, Sandoval F. Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation. Robotics and Autonomous Systems. 2008 Mar 31; 56(3):247-64.