International Journal of Engineering & Technology, 7 (4.27) (2018) 38-43

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Brief Survey on SLAM Methods in Autonomous Vehicle

Talha Takleh Omar Takleh[1]*, Nordin Abu Bakar[2], Shuzlina Abdul Rahman[3], Raseeda Hamzah[4], Zalilah Abd Aziz[5]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Shah Alam, Malaysia.
**Corresponding author E-mail:2016929977@isiswa.uitm.edu.my*

Abstract

The overall purpose of this paper is to provide an introductory survey in the area of Simultaneous Localization and Mapping (SLAM) particularly its utilization in autonomous vehicle or more specifically in self-driving cars, especially after the release of commercial semi-autonomous car like the Tesla vehicles as well as the Google Waymo vehicle. Before we begin diving into the concept of SLAM, we need to understand the importance of SLAM and problems that expand to the various methods developed by numerous researchers to solve it. Thus, in this paper we will start by giving the general concept behind SLAM, followed by sharing details of its different categories and the various methods that form the SLAM function in today's autonomous vehicles; which can solve the SLAM problem. These methods are the current trends that are widely focused in the research community in producing solutions to the SLAM problem; not only in autonomous vehicle but in the robotics field as well. Next, we will compare each of these methods in terms of its pros and cons before concluding the paper by looking at future SLAM challenges.

Keywords: *Bayes' theorem; Extended Kalman Filter; FastSLAM; Graph SLAM.*

1. Introduction

Simultaneous Localization and Mapping (SLAM) is an important area that transcends various fields, particularly in the field on intelligent systems and robotics. The concept of localization deals with the question of where the object is located, while the concept of mapping is to visualize the environment surrounding the object. It is clearly an interesting concept to study because if we can develop and produce SLAM perfectly, the idea of a fully autonomous vehicle be it on the land (self-driving cars), underwater and air, can be accomplished.

However, due to its complexities, there are problems to implement SLAM. In the robotics community, they would refer these complexities as the SLAM problem. The problems have been around for more than 25 years particularly in finding ways or methods to solve it [1][2]. The question is whether an object can be located in an unknown environment, and while the object is situated in that environment, does it have the capability to map its surroundings as well as simultaneously determine its location within that environment. It is this question that many researchers have tried to solve [3]. As a result, there are various methods implemented within SLAM to answer this question.

Although over the years, various methods have been developed to answer the SLAM problem, not all are applicable to self-driving cars. Unlike mobile robots in which majority of the developed SLAM methods were tested on, self-driving cars have bigger parameters to consider, particularly if we want to make it capable of driving autonomously in an urban environment. For example, the area of the environment is larger thus making some of the methods in SLAM not accurate enough, particularly in providing the localization and mapping aspect of the vehicle. Issues such as loop-closure and data association are some of the problematic parameters that arise in more dynamic surroundings, which we can find in an urban environment [1].

There are various methods that have been taken into consideration in solving these issues related to self-driving cars. Global Navigation Satellite System (GNSS) was thought to be able to help deal with the localization aspects [2][9]. Unfortunately, it was deemed problematic due to its signal degradation with respect to the vehicle's location and distance. Another method that has been tested was the Advanced Driver Assistance System (ADAS) that should have helped in the localization of the vehicle. However, not all road information such as lane markings and road edges are available on every road, which make this method ineffective [6][21][22].

Thus, the question is whether the SLAM problem applies for self-driving cars. It should be, although the SLAM problem will require more robust SLAM methods in dealing with the complex and dynamic settings of the urban environment. The SLAM methods for self-driving cars should be able to reduce the uncertainties in finding the localization of the car as well as performing the mapping capabilities as the environment expand.

To answer the problem above, this paper has looked into various researches that utilized numerous methods within SLAM that can solve these problems for self-driving cars. In this paper, we will discuss these methods in greater detail.

In section 2, we will go through the general concept underlying SLAM, such as what constitute the idea of SLAM particularly localization, in the form of Bayes' theorem. We find that this theorem is the main idea behind the majority if not all of the SLAM methods. We will also look into the different categories of SLAM such as filter-based SLAM as well as optimization-based SLAM.

In section 3 we will go into detail of the SLAM methods that have been used to find the solution for the SLAM problem in the research community today. Methods such as Extended Kalman Filter (EKF), FastSLAM and Graph SLAM will be discussed here.

In section 4, we will compare these methods in terms of its pros and cons. We should look into these points in detail to see how it affects the SLAM function for the self-driving cars.

Finally, in section 5, we will conclude this paper by looking back into all of these methods and discuss whether these methods have somehow managed to formulate the solutions to the SLAM problem in self-driving cars. To top it off we will also discuss the future SLAM challenges that could be faced by self-driving cars.

2. SLAM

2.1. Bayes' Theorem

In the SLAM area, there is one theorem or rule that has contribute to the formulation of various important methods in SLAM and that is the Bayes' theorem [10]. These methods are also widely used SLAM methods in autonomous vehicle as we can see later when we go into the details. The Bayes' theorem also gives forth the idea of probabilistic SLAM since through Bayes' theorem important ideas such as Bayes' filter was introduced [3].

In the case of the autonomous vehicle, as the vehicle move from one position to the next, it will continuously take sensor measurements and these sensor measurements can be from a camera (vision-based), laser (LiDAR also known as Light Detection and Ranging) or radar. These measurements can be used in the Bayes' theorem to estimate the probability of the next position of the vehicle with respect to the map. Apart from the sensor measurements, the Bayes' theorem also takes the input from the control inputs such as the wheel encoders.

Fig. 1 gives an example of the LiDAR sensor measurements that can be utilized in the Bayes' theorem as well as any other SLAM methods. The LiDAR sensor measurements were taken using the Velodyne VLP-16 within the compounds of the UiTM campus vicinity.

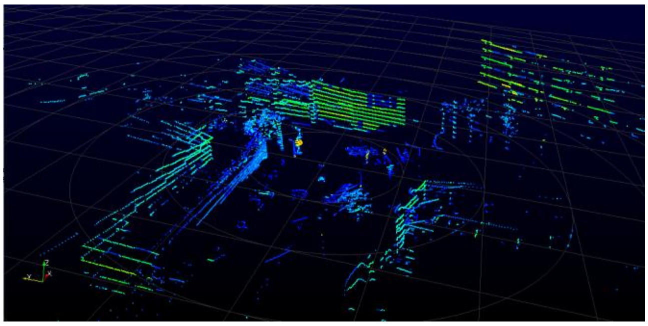

Fig. 1: Example LiDAR sensor measurements

To look at it mathematically, let's declare the variables x_k, k, m, z_k and u_k. The variable x_k is the vehicle pose estimation, k is the time, m is the map of the environment, z_k is the measurements from the sensor readings and u_k is the control inputs from the vehicle. With these variables, the Bayes' rule or theorem can be expressed mathematically as:

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

It is important to take note that on every turn in estimating the probability of the next pose estimation in Bayes' theorem, previous measurements and control inputs need to be taken into consideration. As we can see in (1), $z_{0:k}$ sensor measurements and $u_{0:k}$ input controls were taken from time 0 all the way to time k. This is from the concept of online SLAM that takes estimation measurements in an incremental fashion because it can reduce the computation complexities as the map gets bigger which directly can increase the number of variables [8].

This approach is the total opposite of full SLAM that takes not only the whole sensor measurements and input controls, but as well as the whole vehicle pose estimations, $x_{0:k}$ in its estimation [9][24]. The Bayes' theorem for full SLAM is the same as the online SLAM with the only difference being replacing x_k with $x_{0:k}$.

Thus, with online SLAM, we can use Bayes' theorem to estimate the current position of the vehicle based on the previous sensor and input control measurements, while in the full SLAM we can use Bayes' theorem to estimate the entire trajectory of the vehicle. Fig. 2 shows the relationship of Bayes' theorem with full and online SLAM.

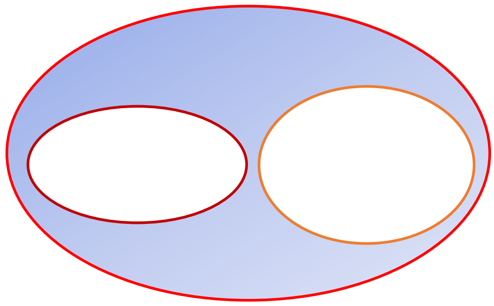

Fig. 2: Graphical representation of Bayes' theorem in SLAM

Under the current self-driving car climate which requires various integration of hardware and software to control them, it is always better to use a system that is not computational heavy since this may produce a number of unwanted problems [1]. These unwanted problems will be discussed further in section 4 when we talked about the pros and cons of the SLAM methods.

In the following part, we will look into the different categories of SLAM and the methods that are based on the Bayes' theorem which have been applied to solve the SLAM problem.

2.2. SLAM Categories

Under the SLAM area there are two categories for the estimation technique [2]. The first is the filter-based and the second is the optimization-based. These are some of the widely used choices that can be made when we plan to solve the SLAM problem.

Fig. 3 shows the different SLAM categories and example methods that can be associated to it.

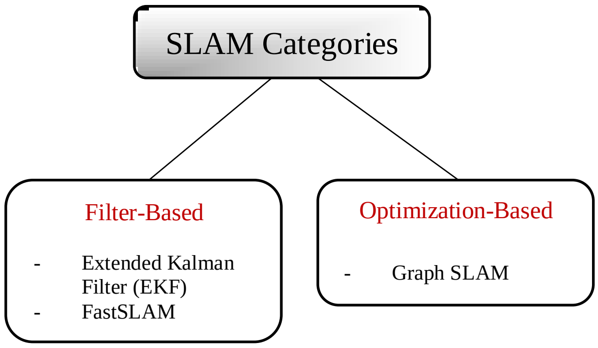

Fig. 3: Different categories for SLAM methods

In the filter-based category, the underlying principle is formed from the Bayes' theorem. It utilizes the Bayesian filter and it functions in a two-step process iteratively [9][10]. The first step is to utilize an evolution model and the control inputs u_k, to make the estimation of the vehicle's pose as well as the map states [10]. The second step is to use the current sensor data measurements z_k and compare it against the map. By carrying out the second step, the filter-based estimation technique can assist in rectifying the wrong prediction, if there is any, that has been acquired from the previous step [10]. These steps will repeat with each new measurement updates. These two steps can be considered as check and balance between an estimation calculation and determine the vehicle's pose position acquired from the sensor data measurements from the map in real time. Example of SLAM methods that utilize the filter-based approach are EKF and FastSLAM.

Similar to the filter-based category, the optimization-based category also has two parts. The first part is to detect the constraints of the problem, in this case the vehicle's current pose, from the sensor data measurements z_k. This can be achieved by finding the connection between the new sensor data measurements and the map [2]. In the second part, we will use the constraints obtained earlier to compute and refine the vehicle's pose as well as previous poses in the map so that we can achieve the unified whole of the estimation of the vehicle's poses with respect to the map. Example of SLAM method that utilizes the optimization-based approach is the Graph SLAM.

These are the two categories in the estimation technique in SLAM. In the next part, we will look at the methods from these categories that have been utilized in solving the SLAM problem in autonomous vehicles as well as in the robotics field.

3. SLAM Methods

3.1. Extended Kalman Filter

The Extended Kalman Filter (EKF) is one of the most widely used filter-based category SLAM method that was utilized to solve the SLAM problem [1][11][13]. The reason why EKF is hugely popular among all the SLAM methods is due to its ability to linearize the non-linear models that are usually faced by autonomous vehicle in the real world.

The underlying concept for the EKF is best described in the survey [3] conducted by Durrant-Whyte and Bailey that described the EKF as the joint posterior distribution taken from the motion and observation model update. To go into more detail, there are two important mathematical formula in EKF. One that describe the motion of the car or better known as the motion model that takes the form of:

$$p(x_k| x_{k-1}, u_k),\ x_k = f(x_{k-1}, u_k) + w_k \tag{2}$$

Where $f(x_{k-1}, u_k)$ describes the vehicle's kinematics and w_k is the additive, zero mean uncorrelated Gaussian noise [11]. By calculating (2), we can estimate the state transition of the car, i.e. the vehicle's location with respect to the control inputs. The other mathematical formula describes the observation model of the vehicle that takes the form of:

$$p(z_k| x_k, m),\ z_k = h(x_k, m) + v_k \tag{3}$$

Where $h(x_k, m)$ describes the geometry of the observation taken from the measurement sensors and v_k is similar to w_k in (2) that is the additive, zero mean uncorrelated Gaussian noise. By calculating (3), we can observe z_k with respect to the location of the vehicle and the landmark location in the map. The use of landmark within the map is important because it can establish loop closure. Together, (2) and (3) can be utilized to compute the mean and covariance of the joint posterior distribution. That itself takes values from the motion and observation-updates.

By calculating the joint posterior distribution, we can establish the vehicle's state or location in the map (based on the landmarks) at time k based on all the sensor measurements $z_{0:k}$ and control inputs $u_{0:k}$. Thus, by calculating the mean and covariance of the joint posterior distribution, the EKF can provide the convergence points with respect to the landmark uncertainty [13].

Therefore, on each motion and measurement-updates, a new joint posterior distribution can be established which can provide the calculation to check the localization of the vehicle with respect to the landmark location within the map. The convergence factor will function as the re-localization of the vehicle towards the corrected landmarks, which should help in establishing the correct loop closure.

3.2. Fast SLAM

FastSLAM is a form of particle filter and can be considered as the second most used filter-based category in SLAM methods utilized to solve the SLAM problem. Apart from its implementation in SLAM, particle filters are also synonymous in path finding and searching methods for various other systems [23][25]. FastSLAM was considered to be the first to model the non-linear processes and does not require Gaussian pose distribution in its method [3][15]. However, the map does have its own set of independent Gaussians.

The underlying concept for FastSLAM has the combination of particle filter, specifically Rao-Blackwellized filter that is responsible in calculating the trajectory with the assistance of weighted samples, and the EKF for landmark estimation [3][5].

In the Rao-Blackwellized filter, a joint distribution at time k - 1 is represented in the set:

$$\{w_{k-1}^{(i)}, x_{k-1}^{(i)}, p(m \mid x_{0:k-1}^{(i)}, z_{0:k-1}^{(i)})\}_i^N \tag{4}$$

Each set has an independent map Gaussian and probability of distribution of the vehicle's state positions and all sensor measurement observations with respect to the landmark in the map. For each particle in (4), a proposal distribution will be computed with respect to the particle history and as a result take a sample from it. This can be achieved through the following equation:

$$x_k^{(i)} \sim \pi\,(x_k \mid x_{0:k-1}^{(i)}, z_{0:k}^{(i)}, u_k) \tag{5}$$

All the samples will be weighted through the importance function that utilizes the independent Gaussian. The importance function utilizes the motion model similar to (2) and the observation model similar to (3) to check the likelihood of the sample with respect to the measurements.

All these steps will be resampled based on the discretion of the user. Some might resample every time-step, after a fixed-number of time-step or after a certain threshold [3]. Finally, the EKF will be performed on every particle. The purpose of this operation is to function as a mapping operation with estimated landmarks obtained through the EKF.

Therefore, with FastSLAM method, we can solve the SLAM problem by sampling the particle element and solve the localization issue with the help of the weighting method that can estimate the trajectory of the vehicle. The FastSLAM also utilizes the EKF to assist in solving the mapping issue.

3.3. Graph SLAM

The Graph SLAM is one of the most widely used SLAM methods from the optimization-based category to solve the SLAM problem. It is considered as the graphical representation of the Bayes' theorem and it utilizes the matrix form in order to correlate the relationship between the landmarks in the map and the vehicle's pose [26][27]. The use of matrix is thought to ease in building the problem relationship and thus utilized in the optimization framework.

The underlying principle of Graph SLAM is to model the SLAM problem particularly the localization issue by modeling it in a matrix that has the information of vehicle states (poses) and landmarks of the map. Once the matrix has been built, we can simply find the minimum of a cost for the trajectory between the vehicle's state and the landmarks. By finding the optimal value, we can estimate the best trajectory for the vehicle within the map [27].

To find the minimum cost, we utilized the function below:

$$F(x, m) = \Sigma_{i,j}\, e_{i,j}(x, m)^T\, \Omega_{i,j}(x, m) \tag{6}$$

In the minimum cost function above, x represents the vector of different vehicle's states (poses) and m represents the landmarks locations within the map. $e_{i,j}$ is the error function that calculates the difference in distance between the estimated states and landmark locations with the observed landmark locations. $\Omega_{i,j}$ is the associated information which contains the vehicle's states (poses) and landmark locations within the map.

Finally, to find the best trajectory for the vehicle within the map, we find the optimal value for the vehicle's state, x^* and the landmarks, m^*. The optimal value is the smallest value from equation (6) and it can be represented in the equation below:

$$(x^*, m^*) = \operatorname{argmin}_{x,m} F(x, m) \quad (7)$$

All three SLAM methods mentioned above have unique characteristics in solving the SLAM problem. The EKF utilizes plenty of mathematical models with the help of Gaussian noise distribution to find the convergence value that can assist in re-localization and mapping, FastSLAM utilizes the concept of sampling and weight function to make the estimation and finally, Graph SLAM utilizes minimum value to find the best trajectory for the vehicle within the map. All three methods, in the end must make sure that the estimated states and landmark locations must be compared with the observation in order to make sure the vehicle move in the correct trajectory.

We have dived deep into these three methods of SLAM that attempt to solve the SLAM problem. It is time to look into these methods and discuss the pros and cons for each of them in the next section.

4. Discussion

As mentioned in the previous section, we have looked into the details of the SLAM methods that try to solve the SLAM problem. However, we need to remind the readers that in most literatures, these methods were either tested on mobile robots or autonomous vehicle, but in a smaller scale. Each of these methods have their own unique characteristics that have been discussed earlier. Although these methods have somewhat proven to be successful in solving the SLAM problem, we need to be aware that in the case of self-driving cars, though similar they might be to mobile robotics in the implementation of SLAM, there might still be other issues that the cars face that mobile robots do not.

In most of the research papers, the testing of these methods is mostly done on a small scale (particularly in the case of autonomous vehicle), thus it is interesting to see if these methods can be robust enough when applied to self-driving cars. In this section, we would like to discuss the pros and cons of each of the methods and discuss whether it is applicable to be implemented in self-driving cars.

4.1. Extended Kalman Filter

The first advantage of EKF is its ability to deal with a non-linear model [4][13]. As mentioned previously, this is extremely important when it comes to self-driving cars because unlike mobile robots, self-driving cars will need to have the ability to deal with the dynamic settings of the real world particularly if it wants to operate in an urban environment. The use of linear models such as the motion and observation model provide the capability for EKF to linearize the non-linear model. This is important in making sure the vehicle is on the right path as it moves along in the map.

Another advantage of EKF is the ability to provide a good reference for loop closure [2][11]. Loop closure provides the ability to recognize if the vehicle has gone through the same landmark in its map. Without it, the estimation of the vehicle's state to landmark will be incorrect and thus can cause some serious trouble for the vehicle.

The final advantage of EKF is that it is a very popular method with large body of work. There are a large number of researches that have been conducted on EKF both on mobile robots and other autonomous vehicle [11][13][14]. Due to its popularity per se, the EKF method has numerous resources for other researchers to learn and build their research. This can be considered as an advantage because for a complex method, the availability of various researches can be beneficial in terms of finding the correct information that can indirectly guide the researchers in the right direction.

Many researches have point out that one of the main problems with EKF is its ineffectiveness when the map becomes larger and more complex. Even the advantage that we have pointed out above has started to be problematic for EKF on a larger and more complex map. One of the main issue is as the map becomes larger and more complex the EKF becomes slower or computationally heavy [3]. This is because of its time complexity that is quadratic in nature as the number of landmarks increases.

The expansion of the map size and complexities also lead to other unsavory characteristics of the EKF. Issues such as problems with data association leads to EKF not being able to associate loop closures properly [3]. Although EKF is known to be able to deal with non-linear models, it is now a cause of concern when it comes to larger and more complex maps. The non-linearity of the environment in these maps can leads to inconsistencies and cause problems in convergence.

4.2. Fast SLAM

The first advantage of FastSLAM is that it can accommodate any distribution, thus it does not require a specific distribution like the Gaussian in the EKF [4][11]. This does not mean it does not utilize the Gaussian distribution at all. As mentioned in the previous section, part of the FastSLAM that utilizes the Rao-Blackwellized filter has its own set of independent Gaussian distribution and this property can increase the speed of the computation [3]. This is highly important in self-driving cars because it gives the ability to make faster decision in the non-linear model.

The use of Rao-Blackwellized filter to sample the vehicle's trajectory has shown that it requires less storage memory due to some particle that will be eliminated in the update process [3]. This can result in a cost-effective process when dealing with the operation of the vehicle.

Since in FastSLAM, each landmark is processed individually through the EKF, it will give the capability to process more landmarks, plus each data association on a per-particle basis [19]. This indirectly will produce a better data association accuracy. Thus, it can reduce the loop closure problem that will be faced by the self-driving cars.

Like any other form of particle filter, FastSLAM does suffer from degeneracy due to its process when sampling the proposal distribution that requires the history of the particle. However, with FastSLAM 2.0, it has been reported that it manages to slow down the rate of degeneracy [20]. Apart from the issue of degeneracy, FastSLAM is also at the disadvantage of sample impoverishment such as particle depletion. However, this issue can also be fixed by avoiding unnecessary sampling steps by resampling according to the Effective Sample Size [20].

4.3. Graph SLAM

The first advantage of Graph SLAM lies within the structure of the matrix that contains the vehicle's pose as well as the landmarks within the map. With this wealth of information an entire trajectory can be visualized, thus this can provide a more accurate and a better consistency in the estimation technique for solving the SLAM problem [27]. Furthermore, the ability of the Graph SLAM to calculate the optimal minimum cost function provide the best possible estimate for the next vehicle's pose to the landmarks, also gives the Graph SLAM a high accuracy and consistency in the estimation technique for solving the SLAM problem. This is important in the case of self-driving cars particularly in dealing with a highly dynamic environment.

The other advantage of Graph SLAM is its ability to consume and process large area of mapping. Again, this is due to its matrix structure that can easily sparse the graph with vehicle's pose and landmark information [27]. This is important especially in self-driving cars that usually operate in a huge area instead of a small controlled area.

In terms of disadvantage, the Graph SLAM does have a high computational cost since it takes into account all the vehicle's pose and landmarks within the map even for a small environment. With the ability to process bigger map, this will only increase the already high computational cost. Thus, this will make Graph SLAM an expensive method to implement. Table 1 provides the summary for the pros and cons of the SLAM methods that have been discussed earlier.

Table 1: Summary of the Pros and Cons of the SLAM Methods

Method	Pros	Cons
EKF	- Able to deal with non-linear model. - Can provide good loop closure reference. - Large body of work.	- Problematic when dealing with large maps that lead to numerous issues.
FastSLAM	- Can accommodate any distribution. - Improve computational speed. - Require less storage memory. - Better data association accuracy	- Sampling data degeneracy issues. - Particle depletion during sampling process.
Graph SLAM	- Can provide better accuracy and consistency in its estimating technique. - Able to consume and process large area of mapping.	- Very high computational cost.

5. Conclusion

In this paper, we presented a brief survey on the SLAM methods that can be utilized to find the solution to the SLAM problem. We looked into the underlying principle behind SLAM; the differences in categories, detail methods as well as the pros and cons. As we examine each method, we found that the EKF, a popular approach to solve the SLAM problem, does not seem suitable for self-driving cars especially in more dynamic and large environment. The FastSLAM seems to be the best method but there were some issues even though it has been partly rectified; plus, there is no concrete literature on how effective it is when it comes to self-driving cars. The Graph SLAM looks promising; however, the high computational cost will be an issue if it was implemented in the self-driving cars. Granted, these are some of the so-called traditional methods that were utilized to solve the SLAM problem, thus it may not fully work for self-driving cars, which require a more robust method. However, given the current literature in SLAM, there are variations to these methods that seems to be more robust. Apart from these variations, there are also new methods that have been researched to solve the SLAM problem. Some of these methods utilize new techniques such as machine learning particularly convolution neural network as we can see in [29] and other techniques like visual SLAM [30]. However, none of these methods are capable of solving the SLAM problem in a definitive manner. Therefore, there are still more opportunities in researching solutions to the SLAM problem.

Acknowledgement

The authors would like to thank Universiti Teknologi MARA, Malaysia and Ministry of Higher Education Malaysia for the facilities and financial support under the national grant 600-IRMI/KPT 5/3/TR (001/2017)-2.

References

[1] C. Cadena, L. Carlone, H. Carillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid and J. J. Leonard," Past, Present and Future of Samultaneous Localization and Mapping: Toward the Robust Perception Age," IEEE Transactions on Robotics, pp. 1309-1332, 2016. Available: IEEE Xplore, https://www.ieee.org. [Accessed April. 23, 2018].

[2] G. Bresson, Z. Alsayed, L. Yu and S. Glaser," Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving," IEEE Transactions on Intelligent Vehicles, pp. 194-220, 2017. Available: IEEE Xplore, https://www.ieee.org. [Accessed April. 23, 2018].

[3] H. Durrant-Whyte and T. Bailey,"Simultaneous localization and map- ping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[4] J. Aulinas, Y. Petillot, J. Salvi, and X. Llado ́, "The SLAM problem: A survey," in *Proc. 11th Int. Conf. Catalan Assoc. Artif. Intell.*, 2008, pp. 363–371.

[5] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108– 117, Sep. 2006.

[6] A. Ranganathan, D. Ilstrup and T. Wu," Lightweight Localization for Vehicles using Road Markings," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 921-927, 2013. Available: IEEE Xplore, https://www.ieee.org. [Accessed April. 25, 2018].

[7] X. Jiang, M. Hoy, H. Yu and J. Dauwels," Linear-complexity Stochastic Variational Bayes Inference for SLAM," IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017. Available: IEEE Xplore, https://www.ieee.org. [Accessed April. 25, 2018].

[8] D. Sun, F. Geiber and B. Nebel," Towards Effective Localization in Dynamic Environments," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4517-4523, 2016. Available: IEEE Xplore, https://www.ieee.org. [Accessed April. 28, 2018].

[9] C. Gentner, T. Jost, W. Wang, S. Zhang, A. Dammann and U.C. Fiebig," Multipath Assisted Positioning with Simultaneous Localization and Mapping," IEEE Transactions on Wireless Communications, pp. 1536-1276, 2016. Available: IEEE Xplore, https://www.ieee.org. [Accessed April. 28, 2018].

[10] J. Mullane, B.N. Vo, M. D. Adams and W. S. Wijesoma," A Random Set Formulation for Bayesian SLAM," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1043-1049, 2008. Available: IEEE Xplore, https://www.ieee.org. [Accessed April. 30, 2018].

[11] X. Xie, Y. Yu, X. Lin and C. Sun," An EKF SLAM Algorithm for Mobile Robot with Sensor Bias Estimation," 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 281-285, 2017. Available: IEEE Xplore, https://www.ieee.org. [Accessed April. 30, 2018].

[12] K. Lee, S. Ryu, J. Kang, H. Choi, C. H. Jun and N. L. Don," Analysis of the Reference Coordinate System Used in the EKF-based SLAM," 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 33-38, 2014. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 2, 2018].

[13] A. A. Ravankar, Y. Kobayashi and T. Emaru," Clustering Based Loop Closure Technique for 2D Robot Mapping Based on EKF-SLAM," 7th Asia Modelling Symposium (AMS), pp. 72-77, 2013. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 2, 2018].

[14] D. Wang, Y. H. Liang, T. Mei, H. Zhu, J. Fu and X. Tao," Lidar Scan Matching EKF-SLAM using the Differential Model of Vehicle Motion," IEEE Intelligent Vehicle Symposium, pp. 908-912, 2013. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 6, 2018].

[15] A. Monjazeb, J. Z. Sasiadek and D. Necsulescu," Autonomous Navigation among large number of nearby landmarks using

FastSLAM and EKF-SLAM – A Comparative Study," 16th International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 369-374, 2011. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 6, 2018].

[16] K. Pakki, B. Chandra, D. W. Gu and I. Postlethwaite," SLAM using EKF, EH∞ and mixed EH2/ H∞ Filter," IEEE International Symposium on Intelligent Control (ISIC), pp. 818-823, 2010. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 6, 2018].

[17] J. Solà," Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrization," IEEE International Symposium on Robotics and Automation (ICRA), pp. 3513-3518, 2010. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 10, 2018].

[18] A. Tamjidi, H. D. Taghirad and A. Aghamohammadi," On the Consistency of EKF-SLAM: Focusing on the Observation Models," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2083-2088, 2009. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 10, 2018].

[19] C. Chen and Y. Cheng," Research of Mobile Robot SLAM based on EKF and its Improved Algorithm," Third International Symposium on Intelligent Information Technology Application (IITA), pp. 548-552, 2009. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 10, 2018].

[20] F. Zhang, S. Li, S. Yuan, E. Sun and L. Zhao," Algorithm analysis of mobile robot SLAM based on Kalman and particle filter," 9th International Conference on Modelling, Identification and Control (ICMIC), pp. 1050-1055, 2017. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 10, 2018].

[21] N. E. Abdul Khalid and M. Manaf," Performance of Optimized Fuzzy Edge Detectors using Particle Swarm Algorithm," Lecture Notes in Computer Science(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 6145 LNCS Part I, pp. 175-182, 2010. Available: UiTM Elsevier, https://uitm.pure.elsevier.com. [Accessed May. 13, 2018].

[22] M. Yusoff, J. Ariffin and A. Mohamed," Discrete Particle Swarm Optimization for Solving a Single to Multistep Destinations in evacuation Planning," 13th International Conference on MAMECTIS, pp. 270-276, 2011. Available: UiTM Elsevier, https://uitm.pure.elsevier.com. [Accessed May. 13, 2018].

[23] S. Masrom, S. Z. Z. Abidin, A. M. Nasir and A. S. Abd Rahman," Hybrid Particle Swarm Optimization for Vehicle Routing Problem with Time Windows," 13th International Conference on MAMECTIS, pp. 142-147, 2011. Available: UiTM Elsevier, https://uitm.pure.elsevier.com. [Accessed May. 13, 2018].

[24] L. Buonocore, S. R. Barros dos Santos, A. A. Neto and C. L. Nacimento," FastSLAM Filter Implementation for Indoor Autonomous Robot," IEEE Intelligent Vehicle Sysmposium (IVS), pp. 484-489, 2016. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 13, 2018].

[25] M. Yusoff, J. Ariffin and A. Mohamed," Solving Vehicle Assignment Problem using Evolutionary Computation," Lecture Notes in Computer Science(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 6145 LNCS Part I, pp. 523-532, 2010. Available: UiTM Elsevier, https://uitm.pure.elsevier.com. [Accessed May. 13, 2018].

[26] I. Deutsch, M. Liu and R. Siegwart," A Framework for Multi-Robot pose Graph SLAM," 18th International Conference on Real-time Computing and Robotics (RCAR), pp. 567-572, 2016. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 15, 2018].

[27] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer Handbook of Robotics*. New York, NY, USA: Springer-Verlag, 2008, pp. 871–889.

[28] R. Wagner, U. Frese and B. Baumi," Graph SLAM with Signed Distance Function Maps on a Humanoid Robot," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2691-2698, 2016. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 17, 2018].

[29] K. Tateno, F. Tombari, I. Laina and N. Navab," CNN-SLAM: Real Time Dense Molecular SLAM with Learned Depth Prediction," 18th International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6565-6574, 2017. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 22, 2018].

[30] M. Chen, S. Yang, X. Yi and D. Wu," Real Time 3D Mapping using a 2D Laser Scanner and IMU-aided visual SLAM," IEEE International Conference on real-time Computing and Robotics (RCAR), pp. 297-302, 2017. Available: IEEE Xplore, https://www.ieee.org. [Accessed May. 22, 2018].