

# Comparative Study of Algorithms for Frontier based Area Exploration and Slam for Mobile Robots

Dayanand V

Department of Computer  
Science and Engg,  
Amrita School of Engg,  
Amrita Vishwa Vidyapeetham  
Coimbatore, India

Rahul Sharma K

Department of Electrical and  
Electronics Engg,  
Amrita School of Engg:  
Amrita Vishwa Vidyapeetham  
Coimbatore, India

Gireesh Kumar T

Center of Cyber Security  
Amrita School of Engg:  
Amrita Vishwa Vidyapeetham  
Coimbatore, India

## ABSTRACT

Exploration strategies are used to guide mobile robots for map building. Usually, exploration strategies work greedily by evaluating a number of candidate observations on the basis of a utility function and selecting the best one. The core challenge in area exploration is to deploy a large number of robots in an unknown environment, map the environment and establishing an efficient communication between the robots. Simultaneous Localization and Mapping (SLAM) comes in to add more accuracy and heuristics to the generic area exploration strategies. Addition to SLAM algorithms will improve the performance of the exploration process and map building to a great extent. In this paper a survey of existing approaches in frontier based area exploration and various SLAM algorithms which can be useful for the process of area exploration are discussed.

## General Terms

Artificial intelligence, Mobile robotics, path planning

## Keywords

Robot navigation, Map building, Frontier based area exploration, SLAM.

## 1. INTRODUCTION

Area exploration is one of the fundamental problems in mobile robotics. Exploring an unknown environment within minimal time is the main goal of any area exploration algorithm. Autonomous area exploration algorithms find applications in space robotics, military operations, disaster management; sensor deployment etc. Area exploration deals with exploring through all unknown areas and creating a map of the environment. Most of the area exploration keeps a map of the environment and updates when an unknown region is explored.

Frontier based area exploration is the most common technique in the field of autonomous area exploration. Frontier means a boundary between known and unknown regions. The basic goal of any frontier based area exploration algorithm is to keep track of all frontiers and the algorithm terminates with the exploration of all frontiers. The early usage of frontier based exploration appeared in Yamuchi [1]. The Frontier based area exploration algorithms can be classified as heuristic based or complete [2]. The various methodologies discussed includes frontier based goal seeking algorithm (FBGS) [2], circle partitioning method, multi-robot exploration using pruning frontiers, MinPos algorithm, and Fast frontier detection (FFD) algorithm. The challenges

involved in the area exploration are navigation, simultaneous localization and mapping, inter- robot communication etc.

The importance of Simultaneous Localization and Mapping (SLAM) algorithms comes in where it improves the accuracy and efficiency of the mapping process making itself useful for the exploration process. SLAM problem in mobile robotics is about building a map of the environment and localizing the robot within the built map [11]. It is very significant since it forms one of the basic problems in motion robotics. Solving the SLAM problem is one of the most noted achievements in robotics research. Within SLAM research, more importance is given towards how the data is modeled, how to handle the association between the data, and how it is represented to improve computational efficiency etc. From the start of interest into SLAM till date, the solutions to SLAM problem is extensively based on probabilistic model. This is because of the realization that SLAM is about finding the conditional probability in a multivariate, multidimensional problem where a high rate of correlation between the data is involved. This paper includes an attempt to list, study and compare some of the mostly used and studied SLAM algorithms in current research.

Remaining sections in this paper are organized as follows: section 2 contains information about the various frontier based goal seeking algorithms and section 3 discusses about the various SLAM algorithms. This is followed by comparative study in section 4 and conclusions in section 5.

## 2. FRONTIER BASED AREA EXPLORATION METHODOLOGIES

### 2.1 Frontier based goal seeking algorithm

Given the probable location of the goal region (GR), i.e. the region where the goal is likely to present, the frontier based goal seeking algorithm [2] aims at reaching the goal. The algorithm aims at proper selection of frontier cells, so that the mobile robot will reach the goal with minimum number of moves. Once GR is reached, the mobile robot is switched to exploration mode to locate the goal. A hexagonal shaped occupancy grid is assumed in this algorithm, so that the robot can reach all its neighbor grids at one step of same length. When the robot starts moving, it senses the environment, and the maps of neighboring cells are updated with the probability of occupancy (Pox). Initially all the cells are set to a prior probability (Pox) of 0.5. Pox of free cells is represented as 0 and that of occupied cells as 1. There are two techniques to route the robot to goal region, Basic Goal Seeking (BGS) algorithm and Modified Goal Seeking (MGS) algorithm.

BGS algorithm chooses the best frontier cell based on goal seeking index and cost function. The cost function to each frontier cells is calculated by considering Pox. A minimum cost path to frontier cell is chosen. Goal seeking index is calculated to choose a frontier cell such that the robot will reach the GR with minimum number of moves. It is calculated by considering the distance between that particular frontier cell and the GR. A variable  $g$  is used to make the robot stay in GR. Another variable  $K1$  is used to switch between goal seeking mode and exploration modes.

MGS algorithm is a modification of BGS algorithm for choosing the frontier cells effectively. A situation called "Trap" is also considered in this algorithm, which is a situation where there are no frontier cells nearby. The algorithm proposes four different strategies to make the robot get out from trap situation.

## 2.2 Area exploration using circle partitioning method

The basic idea of this algorithm is workload sharing. The unknown environment is portioned to sub regions equal to number of robots available by circle partitioning method [3]. Then, robots are assigned with different sub-regions. There are two steps in the algorithm: environment partitioning and sub region coverage. The environment partitioning is done by the method of circle division. The center of area is assumed to be the center of circle and then the circle is divided according to the number of robots available. In sub region coverage, each robot will move to its all frontier cells and continue till no frontier cells remains. In order to cover the whole area, EA\* path planning algorithm is used [4].

## 2.3 Multi-robot exploration using pruning frontiers

In this algorithm the area is divided into sub regions by using K-means clustering algorithm [5]. The assignment of robots to each sub regions to explore the corresponding area is done by Hungarian method [7]. An improvement is also proposed in the algorithm by pruning the frontier cells [6] that reduce the computation time. An occupancy grid (OG) is used in this algorithm, where each cell has a value equal to number of adjacent cells. When a cell is sensed, the value of all adjacent cells is decremented by 1. The area is partitioned into sub regions equal to number of available robots my means of K-means clustering algorithm. The number of clusters is equal to the number of robots and the objects are unknown cells. In robot region assignment, after area partitioning, the algorithm calculates the best assignment of robots to the sub regions. This will make sure that by assignment of robots to the entire area, total cost in sense of robot to region distance is minimized.

The assignment of robots to regions is done by Hungarian method. In this, an ' $n \times n$ ' cost matrix is concerned, where each element represents the cost function of each robot with respect to reach the target point. The distance from robot to each region is calculated using A\* algorithm [4], where robot location is the source and the destination is the region's centroid. This will give an optimal assignment of robots to the regions. The next step is robot-target assignment which will help the robot to choose the frontier cell it should move. This is achieved by calculating the gain function and frontier cell is chosen which has maximum gain. A variable  $\Delta$  is also included in the gain function, which will make sure that the robot will move to a frontier cell in its own assigned region. If robot tends to move to a frontier cell outside its assigned

region, the variable  $\Delta$  will attach a high penalization to the gain function.

An additional consideration is also proposed in the algorithm to decrease the exploration time by pruning the frontier cells that are less favorable. The frontier cells which are very far from the current robot location are less favorable and are pruned from the frontier list by calculating the traveling cost to each frontier cells. Two performance metrics are considered in this algorithm: (1) the total exploration time and (2) total number of movements made.

## 2.4 MinPos: Frontier based area exploration algorithm

This frontier algorithm tries to maintain a balanced spatial distribution of robots in the environment [8]. Each robot evaluates its relative rank among other robots which is dependent on the distance of each robot to frontiers. A frontier will be explored by a robot having lowest rank. The rank is calculated by wave front propagation algorithm (WPA) [9]. Each robot has to perform four steps. (i) Frontier identification and clustering (ii) Computation of distance to frontiers (iii) Assignment to frontiers (iv) Navigation towards the assigned frontiers for a fixed time period. In frontier identification and clustering each robot will maintain a list of all frontier cells. Using wave front propagation algorithm the distance to frontiers is computed. Next the allocation of frontiers to robot is done on the basis of best position of robots to frontiers. Instead of considering distance, position of robot will decide which robots to be assigned to the frontier. The path or trajectory planning of robot to frontiers is performed WPA algorithm.

## 2.5 Fast frontier detection for robotic exploration

Most of the frontier based area exploration algorithms need to have an entire map data during exploration, which is will increase the computation time and memory size requirement. This algorithm proposes two techniques: Wave Front Detector (WFD) algorithm and Fast Frontier Detector (FFD) algorithm [10], which does not need to possess an entire map data. The WFD algorithm is based on breadth first search (BFS) algorithm, which processes the region which is already scanned by the robot and hence no need to scan the entire area. Only known regions are scanned in the BFS algorithm by queuing the known cells in the data structure. The advantage of this algorithm is that the exploration time can be significantly reduced by scanning only known areas.

The FFD algorithm avoids searching both the known and unknown environment, by processing the laser range finder data received in real time. There are basically four steps (i) sorting (ii) contour (iii) detecting new frontiers and (iv) maintaining previously detected frontiers. In sorting stage the laser range finder readings are represented in polar coordinates keeping robot as origin. This reading is then converted to Cartesian coordinated by taking cross product of the vectors. Taking the cross product will help to reduce the computation time compared to the common technique of Euclidean distance estimation. This angle sorted reading is combined to make a complete contour. Then new frontiers are extracted from the previous laser range finder reading. The algorithm will maintain a full list of frontiers which are not explored and the FFD algorithm will be running in background to increase the exploration time.

### 3. SIMULTANEOUS LOCALIZATION AND MAPPING

At each instant  $k$ , the following vectors are considered. Robotic map-building research started 25 years ago, and probabilistic approaches (i.e. Kalman Filters (KF), Particle Filters (PF) and Expectation Maximization (EM)) have become dominant in SLAM since the 1990s. Recursive Bayes rule forms the mathematical base of these three techniques. The main reason for this probabilistic techniques popularity is the fact that robot mapping is characterized by uncertainty and sensor noise. Probabilistic algorithms tackle the problem by modeling different sources of noise and their effects on the measurements [11] explicitly.

#### 3.1 Formulation and Modeling

To follow a consistency among the various representations in SLAM algorithms, we are going to follow the following style of representation in formulation (Fig.1).

At each instant  $k$ , the following vectors are considered.

- $x_k$  is the state vector which contains the position and direction of the robot
- $u_k$  is the control vector which directs the robot to the  $k_{th}$  position from the  $(k - 1)_{th}$  position
- $m_i$  is the position of the  $i_{th}$  landmark
- $z_{ik}$  is the observation of the  $i_{th}$  landmark taken at time  $k$

And,

- $X_{0:k}$  the set of all  $x$ ,  $I : 0$  to  $k_{th}$  instant is the history

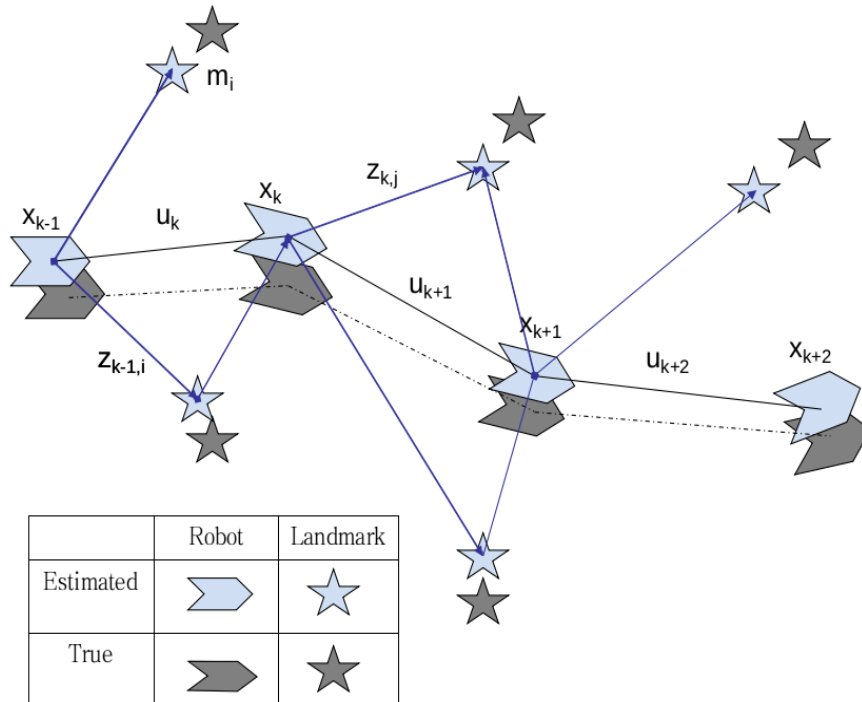


Fig 1: Scenario formulation [12]

#### 3.3 Kalman filter family

Kalman filters are Bayesian filters that represent probabilities  $P(x_t, m | z_t, u_t)$  with Gaussian distributions with small number of parameters. In general terms, Kalman filter is very

of the robot locations

- $U_{0:k}$  that of the control vectors
- $m = \{m_1, m_2, \dots, m_n\}$  is the set of all observed landmarks
- $Z_{0:k} = z_1, z_2, \dots, z_k = Z_{0:k-1}$ ,  $z_k$  is the set of all observations taken of  $m_i$  s.

#### 3.2 Probabilistic SLAM

In probabilistic form, the Simultaneous Localization and Map Building (SLAM)[15] problem requires that the probability distribution  $P(x_k; m | Z_{0:k}, U_{0:k}, x_0)$  be computed for all times  $k$ . Given the recorded observations and control inputs up to and including time  $k$  together with the initial state of the vehicle, the probability distribution gives the joint posterior density of the landmark locations and vehicle state (at time  $k$ ). In general, a recursive solution to the SLAM problem is desirable. When the landmark and vehicle locations are known and generally described in the form  $P(z_k | x_k, m)$ , the observation model describes the probability of making an observation  $z_k$ . The motion model for the vehicle can be described in terms of a probability distribution on state transitions in the form  $P(x_k | x_{k-1}; u_k)$ . That is, the state transition is assumed to be a Markov process in which the next state  $x_k$  depends only on the previous state  $x_{k-1}$  and the applied control  $u_k$ , and is independent of both the observations and the map.

efficient in predicting an estimated state with minimal noise. It can take in inputs which are cluttered by noise and still produce a probabilistically optimal estimate of the state.

The general working of the Kalman filter in a robotic scenario can be summarized as follows. In the first

step, the filter accepts inputs from sensors, and odometer reading and also by considering the current state parameters. Using these inputs, the filter ‘predicts’ estimates of the current state variables (positions), along with their uncertainties (errors). In the ‘correction’ step, the found estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. This procedure is run recursively in real time with the advantage that no additional past information is required except the previous system state.

### 3.3.1 Modeling

Kalman filter model assumes the true state at time  $k$  is evolved from the state at ( $k_1$ ) according to

$$X_k = F_k x_{k-1} + B_k u_k + w_k$$

where,  $F_k$  is the state transition model which is applied to the previous state  $x_{k-1}$ ,  $B_k$  is the control-input model which is applied to the control vector  $u_k$ ,  $w_k$  is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance  $Q_k$ .

### 3.3.2 Extended Kalman Filter (EKF)

Kalman filter works under the assumption that the system is linear dynamic system and that the error measurements follow Gaussian distribution. Extended Kalman Filter (EKF) [12] was developed this shortcoming of the Kalman filter. EKF is a non-linear form of the Kalman filter. The filter linearizes about an estimate of the current mean and covariance. Considering the non-linearity of EKF, the observation model and the state transition model need not be linear here. But it can be differentiable functions.

The overall system in EKF can be expressed mathematically as follows:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}$$

$$z_k = h(x_k) + v_k$$

where  $w_k$  and  $v_k$  are the process and observation noises, which are assumed to follow Gaussian distribution with zero mean. The function  $f$  uses the previous estimate to currently estimate the predicted state and the function  $h$  uses the predicted state to find the predicted measurement. The disadvantages of EKF are that in some cases it does not make a good estimator. If the initial estimate of the state is wrong, the filter may quickly diverge because of the linearization. Also, the estimated covariance matrix tends to underestimate the true covariance matrix.

### 3.3.3 Unscented Kalman Filter (UKF)

Unscented Kalman filter [11] is also a nonlinear Kalman filter and was developed as an improvement to EKF. It is a deterministic sampling technique which uses unscented transforms to pick a minimal set of sample points around the mean. These sample points are called sigma points. Sigma points are then propagated through the non-linear functions, from which the mean and covariance of the estimate are recovered. The advantage of UKF over the normal Kalman filter is that UKF is able to accurately capture the true mean and covariance.

## 3.4 Particle filter

Particle filters are sequential Monte-Carlo methods. It is a recursive Bayesian filter. The sequential Monte-Carlo estimation is done by using set of random points as clusters/particles representing the Bayesian predicted posterior probability. Particle filters are generally not suited

for real time applications since the computational complexity will become much higher at higher dimensions [14].

A set of samples taken from the posterior distribution are used by particle filters to represent any distribution given as input and will be non-parametric in nature. Using the measurement model, weights are given to the particles. According to the posterior the samples are redistributed through re-sampling. The working process of particle filters in a real time scenario can be summarized as follows. Initially, based on the individual weights, a fixed number of particles are selected. Some particle, if important to the scenario may be picked up multiple times. Then, the particles are moved according to the deterministic dynamics. Now, for each sample, a prediction and comparison is done to get likelihood. Finally, the weights of individual particles are updated according to the measured likelihood value. This process is repeated until the goal is obtained.

### 3.4.1 Rao-Blackwellized particle

Rao-Blackwellized particle filter [12] [16] is an extension of normal particle filters by application of Rao-Blackwell theorem [17]. The aim is to estimate

$$P(x_{1:t}, m | z_{1:t}, u_{1:t} - 1)$$

where  $m$  is the map and  $x_{1:t}$  is the trajectory of robot.

This probability value is factorized to the product of two terms as,

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

There are four steps which follow. They are sampling, importance weighting, re-sampling and map estimation. Sampling involves obtaining the next generation  $x_t$  of the particles from the generation  $x_{t-1}$  from the proposal distribution. Next step is importance weighting in which an importance weight  $w_t^i$  is assigned to each particle as:

$$w_t^i = \frac{P(x_{1:t}^i | z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}^i | z_{1:t}, u_{1:t-1})}$$

In the re-sampling step, particles are drawn with replacement proportional to their importance weight. Finally in the map estimation step, for each particle, the corresponding map estimate  $P(m(i) | x^i, z_{1:t})$  is computed based on the trajectory  $x_{1:t}^i$  of that sample and the history of observations  $z_{1:t}$ .

## 3.5 Expectation Maximization based methods

Expectation Maximization based methods [11] is based on the context of maximum likelihood (ML) estimation. EM methods offer an optimal solution making it an ideal option for map building. But it does not show that effectiveness for localization.

The two steps involved in EM methods are:-

1. Expectation (E-Step): The posterior over robot poses is calculated for a given map
2. Maximization step (M-step): The most likely map is calculated given these pose expectations

EM methods tackle the data association problem well by localizing repeatedly in E-step. In the M-step, these correspondences are translated into features in the map. This feature then either gets reinforced in the next E-step, or they gradually disappear. The same data gets processed many

times which will induce inefficiency for real time applications. Therefore EM is usually combined with PF (M step to construct map, PF to localize within the map) for usage in SLAM purposes.

#### 4. COMPARITIVE STUDIES

Each algorithm performs best in certain environments. A comparison of the various area exploration algorithms is done. A comparison is shown in Table 1.  
A comparison of some of the SLAM algorithms discussed earlier is listed in Table 2.

#### 5. CONCLUSION

Frontier based area exploration is the most commonly used technique for area exploration problem. State-of-art frontier based algorithm process the entire map and if the area

exploration region is a large region, the robot will have to wait for a long duration until the algorithm terminates. The recent algorithms propose different techniques to make the area exploration faster. If only a goal region is known and not the goal, BGS and MGS algorithms performs best. The circle portioning algorithm and pruning algorithm aims at work load sharing and faster area exploration. MinPos algorithm proposes a distributed, less complex technique for area exploration by maintaining a good spatial distribution of robots. Fast frontier based algorithm does not process entire map, instead it process the data from the contour it made with the help of laser range finder. A study of various SLAM algorithms is done. Each SLAM algorithm is observed to be the best in its corresponding situation as observed from Table 2.

**Table 1. Comparison between different area exploration algorithms**

	BGS and MGS	Circle portioning	Pruning	MinPos	Fast Frontier
Environment representation	Hexagonal OG	Circle partitioning and OG	OG	OG	Contour
Basic Algorithm	BGD and MGS	Circle partitioning	K-means clustering, Hungarian Method and pruning technique	MinPos	WFD and FFD
Path planning algorithm	Using goal seeking index	EA*	A*	WPA	BFS

**Table 2. Comparison study of various SLAM algorithms**

	Kalman filter family	Particle filters	Expectation Maximization method
Pros	1.High convergence 2.Handles uncertainty	1.Handles non-linearity 2.Handles non-Gaussian noise 3.Global localization is possible	1.Optimal for map building
Cons	1.Gaussian assumption 2.Slow in high dimensional maps 3.Global localization is not possible	1.Growth in complexity	1.Inefficient, cost growth 2.Unstable for large scenarios 3.Only successful in map building

Circle Partitioning Method. Procedia Engineering 41 (2012) 383-387

#### 6. REFERENCES

- [1] Yamauchi, Brian : Frontier-based exploration using multiple robots. In proceeding of the second International Conference on Autonomous agents (1998)47-53
- [2] Jisha, V. R., and Debasish Ghose : Goal seeking for robots in unknown environments. In Intelligent Robots and Systems (IROS), 2010 International Conference on IEEE (2010)4692-4697
- [3] Jain, Upma, Ritu Tiwari, Samriddhi Majumdar, and Sanjeev Sharma: Multi Robot Area Exploration Using
- [4] Pal, Anshika, Ritu Tiwari, and Anupam Shukla: Multi Robot Exploration Using a Modified A\* Algorithm. In Intelligent Information and Database Systems, Springer Berlin Heidelberg, (2011) 506-516.
- [5] A. Solanas, M. A. Garcia: Coordinated multi-robot exploration through unsupervised clustering of unknown space. Proc. Int. Conf. on Intelligent Robots and Systems, vol.1 (2004) 717-721

- [6] Pal, Anshika, Ritu Tiwari, and Anupam Shukla: Multi robot exploration through pruning frontiers. *Advanced Materials Research* 462 (2012) 609-616
- [7] H.W. Kuhn: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1) (1955) 83–97
- [8] Bautin, Antoine, Olivier Simonin, and François Charpillet: MinPos: a novel frontier allocation algorithm for multi-robot exploration. In *Intelligent Robotics and Applications*, Springer Berlin Heidelberg (2012) 496-508
- [9] Adel Al-Jumaily, Cindy Leung: Wavefront Propagation and Fuzzy Based Autonomous Navigation. *International Journal of Advanced Robotic Systems*, Volume 2, Number 2 (2005) 93-102
- [10] Keidar, Matan, Eran Sadeh-Or, and Gal A. Kaminka: Fast frontier detection for robot exploration. In *Advanced Agent Technology*, Springer Berlin Heidelberg (2012) 281-294
- [11] Josep Aulians., Yvan Petillot., Joaquim Salvi., and Xavier Llado.: The SLAM problem: a survey, *Proceeding of the 2008 conference on Artificial Intelligence Research and Development* (2008) 363-371.
- [12] Hugh Durrant-Whyte., and Tim Bailey.: Simultaneous Localization and Mapping: Part I, *IEEE Robotics and Automation Magazine* (June 2006) 99-108.
- [13] Hugh Durrant-Whyte., and Tim Bailey.: Simultaneous Localization and Mapping (SLAM): Part II, *IEEE Robotics and Automation Magazine* (September 2006) 108-117.
- [14] Sebastian Thrun., CA Wolfram., Dieter Fox., *Probabilistic Robotics*.
- [15] Soren Riisgaard and Morten Rufus Blas.: SLAM for Dummies: A Tutorial Approach to Simultaneous Localization and Mapping.
- [16] Nosan Kwak., Kazuhito Yokoi., and Beom-Hee Lee.: Analysis of Rank-Based Resampling Based on Particle Diversity in the Rao–Blackwellized Particle Filter for Simultaneous Localization and Mapping, *Advanced Robotics* 24 (2010) 585–604.
- [17] Casella G., and Robert C. P.: Rao-Blackwellisation of sampling schemes, *Biometrika* (1996) 83(1), 81-94.4