

Article

Reliable and Fast Localization in Ambiguous Environments Using Ambiguity Grid Map

Gen Li, Jie Meng, Yuanlong Xie, Xiaolong Zhang, Yu Huang *, Liquan Jiang and Chao Liu

School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

* Correspondence: yuhuang_hust@hust.edu.cn; Tel.: +86-027-87557765

Received: 28 June 2019; Accepted: 26 July 2019; Published: 29 July 2019



Abstract: In real-world robotic navigation, some ambiguous environments contain symmetrical or featureless areas that may cause the perceptual aliasing of external sensors. As a result of that, the uncorrected localization errors will accumulate during the localization process, which imposes difficulties to locate a robot in such a situation. Using the ambiguity grid map (AGM), we address this problem by proposing a novel probabilistic localization method, referred to as AGM-based adaptive Monte Carlo localization. AGM has the capacity of evaluating the environmental ambiguity with average ambiguity error and estimating the possible localization error at a given pose. Benefiting from the constructed AGM, our localization method is derived from an improved Dynamic Bayes network to reason about the robot's pose as well as the accumulated localization error. Moreover, a portal motion model is presented to achieve more reliable pose prediction without time-consuming implementation, and thus the accumulated localization error can be corrected immediately when the robot moving through an ambiguous area. Simulation and real-world experiments demonstrate that the proposed method improves localization reliability while maintains efficiency in ambiguous environments.

Keywords: navigation; perceptual aliasing; ambiguous environment; Monte Carlo localization; Dynamic Bayes network

1. Introduction

For an autonomous robot, reliable and fast localization is a critical prerequisite during navigation. However, ambiguous environments may cause perception aliasing of the external sensors [1,2], which makes it a challenging task for a robot to accurately determine its pose or even leads to localization failure. As demonstrated in Figure 1, the ambiguous environments are specified as the environments that contain ambiguous areas, including long corridors, empty space, and tangled thicket. In such environments, however, there still exist some unambiguous areas that can help the robot to locate itself. For this end, constructing a localization method on the basis of the modeled ambiguity of the environment can increase the localization reliability without loss of efficiency.

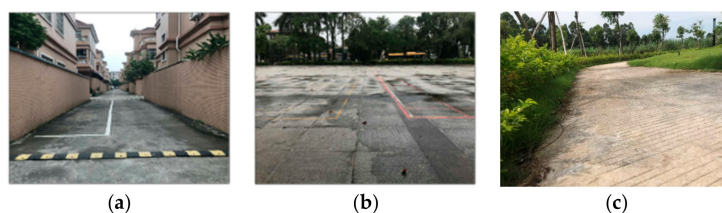


Figure 1. Ambiguous environments: (a) long corridor; (b) empty space; (c) tangled thicket.

For localization procedure of a robot, global localization is firstly performed to obtain the initial pose by using the observation data from external sensors, followed by the pose tracking which enables

the robot to rapidly track its pose [3]. To be more specific, pose tracking consists of two steps, i.e., prediction and update. In the prediction step, based on the odometry motion model, the pose increment obtained by internal sensors is incorporated into the results of the previous localization moment to predict the current pose. As for the update, the localization error of the predicted pose is corrected by the observation data of external sensors, which respects the observation model to produce the final localization results of the current moment.

Recently, most of the existing localization methods [4–9] only rely on the external sensors in the update step, and it remains a tough job to satisfy the practical requirements when rectifying the localization error in an ambiguous environment. This is because the external sensors may suffer from perceptual aliasing, which implies that the observation data captured at different pose are difficult to distinguish due to the environmental ambiguity. Consequently, the localization error of the prediction step may accumulate as the robot moves through an ambiguous area of the environment. Even if the robot reaches an unambiguous area, the accumulated localization error is often too large to be corrected by those methods.

To solve the localization problem in an ambiguous environment, additional facilities such as artificial reflectors [10,11] and wireless sensor networks [12] have attracted increasing attention. With additional facilities, these methods require additional maintenance costs and need to modify the existing environment, which are not unreasonable for some practical usage. Assuming that a robot will leave the area where perceptual aliasing occurs and enters an unambiguous area, the pose of the robot can be recovered by a global localization method. In an unambiguous area, although global localization method can be used to estimate the pose of a robot according to the observation data captured by external sensors, the operating time of global localization is much longer than that of pose tracking. In addition, it is necessary to solve the detection problem of when the robot enters the unambiguous area. Additional information [13,14] except for the readings of internal sensors is utilized in the prediction step of pose tracking to allow more reliable pose prediction. However, their methods are impractical to solve the robot localization problem in ambiguous environments due to the fact that they do not explicitly consider the ambiguity of an environment and such environment property contains useful information for robot localization.

For robot navigation, the adaptive Monte Carlo localization (AMCL) method is able to achieve effective and fast robot localization in different environments [6–9] and the particle representation used in AMCL has included some effects of ambiguity. However, due to the limitation of particle number, the inaccurate motion model and observation model, it cannot cope with long term perception aliasing. In this paper, the AGM which explicitly models the ambiguity of an environment is utilized to improve the standard AMCL so that it can correct the accumulated localization error after robot moving through an ambiguous area and ultimately achieve a fast and reliable robot localization. The main contributions of this paper are summarized as follows:

- The AGM is achieved by proposing a new localizability evaluation method called average ambiguity error (AAE) so that the possible localization error at a given pose can be estimated, and those unambiguous areas of an environment can be identified.
- By integrating the AGM, the standard Dynamic Bayes network (DBN) for robot localization is improved to model the localization problem in ambiguous environments. Moreover, a new motion model referred to as portal motion model is implemented to obtain more reliable pose prediction in ambiguous areas.
- Based on the improved DBN, the AGM-based adaptive Monte Carlo localization (AGM-AMCL) method is derived to achieve fast probability inference.
- Simulation and real-world experiments validate the effectiveness of the AGM and the AGM-AMCL method, which can reliably locate a robot with guaranteed efficiency in three different ambiguous environments.

The remainder of this paper is constructed as follows: Section 2 presents a range of related work in terms of localizability evaluation and reliable localization. Section 3 provides the AGM and its detailed implementation. In Section 4, we describe the improved DBN, and then we derive the portal motion model and the AGM-AMCL. Finally, in Section 5, simulations and real-world experiments are conducted to evaluate the performance of the AGM and the AGM-AMCL method. Finally, some conclusions are offered in Section 6.

2. Related Work

Considering the key issue we attempt to solve in this paper, we will focus on the related work of localizability evaluation and reliable localization in ambiguous environments. To the best of our knowledge, there are few articles explicitly dealing with the localization problem in ambiguous environments. Global localization methods and optimized pose tracking methods can be used to relocate a robot after the robot leaving an ambiguous area and entering an unambiguous area. Therefore those methods were introduced in the reliable localization part.

2.1. Localizability Evaluation

Evaluating the localizability in an environment is to quantitatively estimate the achievable localization accuracy of a given pose. In general, localizability evaluation methods are not related to the property of an environment, therefore they are applicable in both ambiguous environments and unambiguous environments. Fisher information matrix (FIM) is a useful indicator for evaluating environmental localizability. Based on the Cramér-Rao lower bound, Censi [15,16] used FIM to represent the theoretical accuracy limit of a localization method employing range-finder data and attempted to use FIM to analyze the achievable accuracy of pose tracking. FIM has no analytical solution if an environment is represented by the occupancy grid map. To solve this problem, a discrete FIM [17] was proposed, which was obtained by simulated scan rays from an occupancy grid map. Afterwards, Irani et al. [18] presented a path planning method for a mobile manipulator based on discrete FIM. The localizability evaluation methods based on FIM can be used to evaluate the theoretical localization accuracy limit of a certain position, however, it is difficult for FIM to estimate the localization accuracy in a real localization process. This is because, under a given environment, the FIM is only related to the robot's pose, without considering the environmental similarity of different poses and the observation model used in the actual localization process.

In addition to the localizability evaluation based on the FIM, there have been some related studies in recent years. In [19], a novel strategy for incremental sampling-based motion planning was proposed, which introduced a new evaluation of localizability using the trace change rate of covariance matrix from the prediction step to the update step of an extended Kalman filter. Zhen et al. [20] proposed a method to evaluate the localizability based on the constraints provided by a sensor measurement. The above methods can evaluate the environmental localizability relatively while the evaluation value itself is meaningless for estimating localization error. In [21], to evaluate the localizability, a vehicle was driven through the designated environment and an error model was learned. However, their method takes a long time to experiment and is not suitable for large scale environments. Javanmardi et al. [22] adopted four types of criteria to evaluate the capability of maps for localization. Based on their criteria, the localizability of a map can be comprehensively evaluated. Our method differs from theirs in the sense that the observation model adopted in the real localization process and the sensor measurement uncertainties are involved in our evaluate method.

2.2. Reliable Localization

In recent years, many localization methods have been proposed to enhance localization reliability. One of the commonly used methods is adding additional facilities. Our previous work [10] proposed a new observation model for MCL, which improves the robustness of localization by reliable reflector prediction in the ambiguous environments. Beinhofer et al. [23] addressed the data association problem

by deploying a limited number of uniquely identifiable artificial landmarks. A practical approach to landmark placement was presented in [24] that minimizes the localization errors. Besides, wireless sensor networks [12], radio frequency identification [25,26], etc. were also applied to achieve reliable localization. However, the additional facilities are costly for placement and maintenance, which limits the abovementioned method for potential applications.

Once localization fails, re-localization can be achieved through global localization, thereby improving the reliability of localization. A particle filter improved by a particle swarm optimization algorithm was proposed in [27]. With their method, more robust global localization can be achieved using fewer particles. A laser scan matching with an improved probabilistically-motivated Maximum Likelihood Estimation algorithm is adopted to obtain the global optimum position [2], which can reach an updating rate of 2 Hz in a feature-poor environment. A novel branch-and-bound multiresolution global localization method using a multiresolution Gaussian mixture map was proposed in [28] which was used for real-time localization in urban environments. By finding correspondences between a set of sparse features detected in two occupancy grid maps, a map to map matching method was designed [29], which can be used for global localization. In [30], a differential evolution approach was tested for global localization in an environment with different types of occlusions, which used three different cost functions based on a non-symmetric fitness function. The above methods attempt to achieve reliable and fast global localization. However, due to the need for global optimization, their running time is still longer than pose tracking. In [31] and [32], classifiers for pose recognition were trained by the machine learning method. Through such classifiers, the global pose of a robot can be identified quickly. However, those methods rely on reliable feature extraction methods, and wrong feature extraction may result in large localization errors.

By optimizing the prediction step of pose tracking, it can generate more accurate pose samples, also called particles, and improve the reliability of localization. Gutmann et al. [33] used sensor resetting method to generate some pose particles in the prediction step. The number of particles relies on the ratio between the short term and the long term average observation likelihood. Building on the basic Monte Carlo localization algorithm, a robust localization algorithm called Mixture-MCL was developed [34], which attempted to generate pose samples according to the observation model using a set of learned kd-trees in the prediction step to complement the odometry motion model. However, this method takes a large amount of memory, and as the size of the environment grows, the amount of memory required grows rapidly. Zhang et al. [13] proposed the concept of similar energy region. Extra pose samples can be generated in similar energy region to obtain a reliable pose prediction. Oh et al. [14] developed an Augmented Motion model, which used the semantic information available in maps to bias the motion model toward areas of higher probability. In [35], by using a global hypothetical pose set to represent an environment, a novel qualitative motion model is proposed to generate extra pose hypothesis. In [13,14,33–35], additional information is used in the prediction step to generate extra pose samples which may be closer to the robot true pose. Once the accumulated localization error is too large, extra pose samples may compensate for such error and improve the reliability of localization. However, their methods do not explicitly consider the ambiguous nature of an environment, it is very difficult for these methods to achieve reliable localization in ambiguous environments. Inspired by the abovementioned methods, additional information from the AGM is exploited to improve the prediction step of the AMCL method, which allows our method to generate some extra particles at a proper time with proper distribution and effectively improves the reliability of localization.

3. AGM

AGM is used for modeling the ambiguity of an environment, such that we can figure out which positions suffer from perceptual aliasing and how much localization error will be caused by perceptual aliasing at those positions. Analogical to the occupancy grid map, AGM has multiple cells, each of which measures the localization error resulting from perceptual aliasing. In this paper, we refer the

aforesaid localization error to the AAE, and it is crucial to build the AGM. Practically, it is noticed that a robot cannot distinguish two different poses effectively due to the similar surroundings of these poses or the considerable sensor noise or poor observation model [36]. Considering that, the formulation of AAE should be related to the environment as well as the external sensor and the constructed observation model.

Define $\mathbf{x} = (\mathbf{p}, \theta)$ as a pose with $\mathbf{p} = (x, y)$ used to denote a position in two-dimension. Let \mathbf{z}_x be the observation data obtained by the external sensor of a robot at the pose \mathbf{x} . Using the observation data \mathbf{z}_x and an input pose \mathbf{y} , the likelihood function of an observation model is denoted by:

$$L_x(\mathbf{y}) \triangleq P(\mathbf{z}_x|\mathbf{y}) \quad (1)$$

From Equation (1), we conclude that $L_x(\mathbf{x})$ represents the likelihood value when the input pose \mathbf{y} is equal to the robot's true pose. For a general pose \mathbf{y} if the maximum value of $L_x(\mathbf{y})$ is identical with $L_x(\mathbf{x})$, the robot's pose can be accurately obtained through maximum likelihood estimation. However, in an ambiguous environment, there are other values of \mathbf{y} that makes $L_x(\mathbf{y})$ equal or bigger than $L_x(\mathbf{x})$. In this situation, it is difficult to distinguish pose \mathbf{x} and pose \mathbf{y} based on this likelihood function. Thus, in this paper, the probability of a robot's true pose \mathbf{x} confused with some other pose \mathbf{y} can be defined as:

$$P_{y|x}^A \triangleq P(L_x(\mathbf{y}) + \varepsilon \geq L_x(\mathbf{x})) \quad (2)$$

where ε is a small threshold to compensate the unmodeled factors of the likelihood computation.

Given the observation data \mathbf{z}_x , the likelihood value of $L_x(\mathbf{y})$ can be uniquely determined. Thus probability $P_{y|x}^A$ is rely on the distribution of \mathbf{z}_x and Equation (2) can be expressed as:

$$P_{y|x}^A = \int_{\mathbf{z}_x} \mathbf{1}_{(L_x(\mathbf{y}) + \varepsilon \geq L_x(\mathbf{x})|\mathbf{z}_x)}(\mathbf{z}_x) P(\mathbf{z}_x) d\mathbf{z}_x \quad (3)$$

where $P(\mathbf{z}_x)$ is the distribution of observation data at the pose \mathbf{x} and $\mathbf{1}_{(L_x(\mathbf{y}) + \varepsilon \geq L_x(\mathbf{x})|\mathbf{z}_x)}(\mathbf{z}_x)$ denote an indicator function, i.e.:

$$\mathbf{1}_{(L_x(\mathbf{y}) + \varepsilon \geq L_x(\mathbf{x})|\mathbf{z}_x)}(\mathbf{z}_x) = \begin{cases} 1 & L_x(\mathbf{y}) + \varepsilon \geq L_x(\mathbf{x})|\mathbf{z}_x \\ 0 & L_x(\mathbf{y}) + \varepsilon < L_x(\mathbf{x})|\mathbf{z}_x \end{cases} \quad (4)$$

Since $P_{y|x}^A$ cannot be determined analytically, we employ Monte Carlo integral to achieve its approximation as:

$$P_{y|x}^A \approx \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{(L_x(\mathbf{y}) + \varepsilon \geq L_x(\mathbf{x})|z_x^i)}(z_x^i) \quad (5)$$

where z_x^i denotes the i -th sample of the observation data, N denotes the total number of samples.

Now we define the AAE of a given pose \mathbf{x} as:

$$aae(\mathbf{x}) = \frac{\sum_{\Delta_i \in M} d(\mathbf{x} + \Delta_i, \mathbf{x}) P_{\mathbf{x} + \Delta_i | \mathbf{x}}^A}{\sum_{\Delta_i \in M} P_{\mathbf{x} + \Delta_i | \mathbf{x}}^A} \quad (6)$$

where M is a set of the incremental pose, $\Delta_i \triangleq (\Delta x_i, \Delta y_i, \Delta \theta_i)$ denotes the i -th element in M , $d(\mathbf{x} + \Delta_i, \mathbf{x})$ denotes the difference between $\mathbf{x} + \Delta_i$ and \mathbf{x} , which is expressed by:

$$d(\mathbf{x} + \Delta_i, \mathbf{x}) = \sqrt{\Delta x_i^2 + \Delta y_i^2} + \xi |\Delta \theta_i| \quad (7)$$

where ξ is a coefficient factor used to convert the orientation difference into the distance difference.

From Equation (6), it is known that the AAE of a pose represents the weighted average localization error caused by the ambiguity between the true pose and the other poses which are contained by M . Large AAE value of a given pose means that a large localization error may be induced after a robot moving through that pose. Moreover, by exploiting the AAE value of a trajectory, the accumulated localization can be estimated in some way and this property is used in our localization method in the following section.

Then, based on Equation (6), the AAE of a position can be derived directly:

$$aae(\mathbf{p}_x) = \frac{1}{n_\theta} \sum_{\theta=0^\circ}^{360^\circ} aae(\mathbf{x}(\mathbf{p}, \theta)) \quad (8)$$

where n_θ denotes the number of discrete θ .

Figure 2 shows a schematic diagram of M . Defining $R_d(M)$ and $R_\theta(M)$ as the distance range and the orientations range of the set M , respectively, we have:

$$M = \left\{ \Delta_i \triangleq (\Delta x_i, \Delta y_i, \Delta \theta_i) \mid \sqrt{\Delta x_i^2 + \Delta y_i^2} \leq R_d(M) \wedge |\Delta \theta_i| \leq R_\theta(M) \right\} \quad (9)$$

In this paper, in accordance with occupancy grid map, Δx_i and Δy_i are pre-selected as an integer multiple of the grid map resolution and $\Delta \theta_i$ is selected as an integer multiple of 3° to make the trade-off between the efficiency and accuracy.

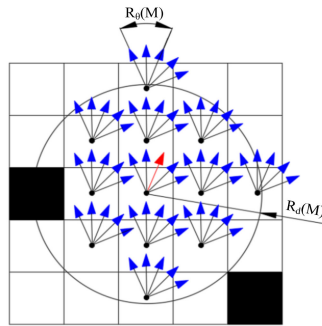


Figure 2. Schematic diagram of M . The black grids and white grids indicate the occupied cells and free space cells, separately. The $R_d(M)$ and $R_\theta(M)$ are marked on this figure. The position and the direction of the red arrow represent the pose x to be evaluated. Those blue arrows represent those poses derived from the incremental pose set M and x according to $x + \Delta_i$.

To derive a solvable solution for AAE determined by Equation (8), we need to focus on the sampling method of $P(z_x)$ and the configuration of ξ , $R_d(M)$ and $R_\theta(M)$. In addition, considering that a 2-D laser range finder is used as the external sensor in this paper, the following method for computing $P(z_x)$ is based on the 2-D laser range finder. First, sampling with a large amount of data is time-consuming and unrealistic in a large-scale environment. As studied in [37], the 2-D laser range finder and the beam model are suggested to be used as the external sensor and observation model for the practical implementation, respectively. Then we can adopt the following feasible and effective method to obtain observation data samples: (1) build the occupancy grid map of an environment using SLAM method; (2) determine the parameters of the beam model by real observation data using learning method presented in [37]; (3) obtain the observation data samples at various poses through simulation. Second, the ξ is expressed by $\xi = \chi_1 / \text{rad}(\chi_2)$, where χ_1 and χ_2 denotes the maximum tolerable distance error and the maximum tolerable orientation error for reliable localization, separately. Finally, by using the inequality $aae(x_p) \leq R_d(M) + \xi R_\theta(M)$, which can be derived from Equations (6) and (7), we can obtain the range constraints for M , i.e., $\chi_1 \leq R_d(M)$, $\chi_2 \leq R_\theta(M)$. More accuracy

localizability evaluation can be obtained through a larger range of M while it is time-consuming. Therefore, we select $R_d(M) = \chi_1$ and $R_\theta(M) = \chi_2$.

4. AGM-AMCL

As shown in Figure 3, the standard AMCL [38] mainly contains three steps: resampling step, prediction step, and update step. In the resampling step, particles are sampled according to the likelihood of each particle. Then new particles are generated in the prediction step based on the odometry motion model. After that, the likelihood of each particle is evaluated by the observation model in the update step. In addition to those three steps, our method also includes a step for estimating the accumulated ambiguity error using the AGM. Moreover, in the prediction step of our method, the portal motion model is used to generate extra particles in an adjacent unambiguous area if a robot is in an ambiguous area. Due to the extra particles, large localization error can be corrected as soon as the robot enters the adjacent unambiguous areas.

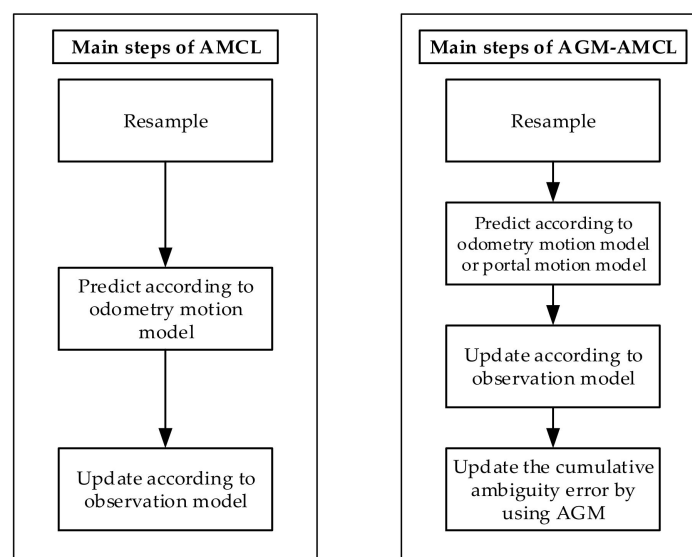


Figure 3. Main steps of AMCL and our method.

In fact, our method can be derived from a DBN which models the robot localization problem in unambiguous environments. In the following part, the derivation of our method, as well as the portal motion model and the estimation method for the accumulated localization error using the AGM, will be detailed introduced.

4.1. Standard DBN for Localization

Given a prior occupancy grid map m_g , the robot localization can be modeled as a probabilistic inference problem:

$$Bel(x_t) = P(x_t | z_{1:t}, u_{1:t}, m_g) \quad (10)$$

where x_t denotes the robot pose at a time t , $z_{1:t}$ is the sequence of observations, $u_{1:t}$ denotes a sequence of motion control commands or odometry measurements.

Figure 4 demonstrates the standard DBN [39] for robot localization which contains two importance independent assumptions: Markov independence of the odometry and Markov independence of the observation.

According to the independent assumptions embedded in the standard DBN, we can simplify the robot pose belief that presented in Equation (10) as:

$$Bel(x_t) = \eta P(z_t|x_t, m_g) \int_{x_{t-1}} P(x_t|x_{t-1}, u_t) Bel(x_{t-1}) dx_{t-1} \tag{11}$$

where η is a normalization coefficient, $P(x_t|x_{t-1}, u_t)$ and $P(z_t|x_t, m_g)$ are corresponding to the motion model and observation model, respectively. To fast infer the robot's pose belief in Equation (11), AMCL [38] can be used with the KLD-sampling method regulating the particle set size.

Because of perception aliasing, the localization error accumulates rapidly when a robot moves through an ambiguous area. However, the DBN, shown in Figure 4, does not explicitly consider the accumulated localization error caused by perception aliasing so that it is not suitable for modeling the robot localization problem in ambiguous environments. For this reason, an improved DBN is proposed.

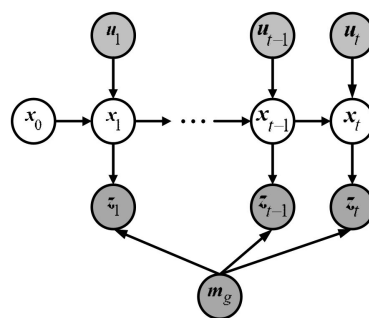


Figure 4. Standard DBN for localization problem.

4.2. Improved DBN

The improved DBN is shown in Figure 5, where m_g stands for the occupancy grid map and m_a is the corresponding AGM, e_t is the cumulative ambiguity error at time t , which reflects the accumulated localization error caused by perception aliasing.

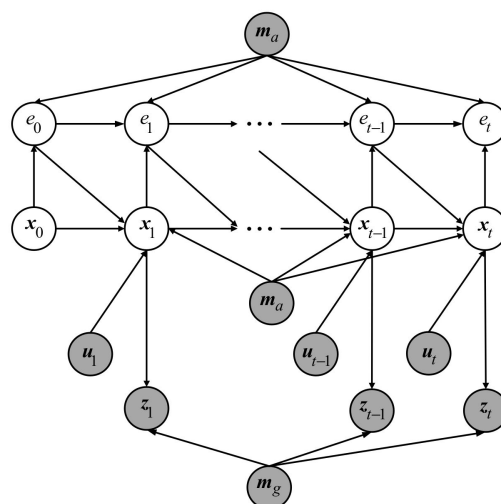


Figure 5. Improved DBN for localization problem. The AGM node (m_a) and the cumulative ambiguity error node (e_t) are introduced.

To infer the robot's pose, as well as the accumulated localization error, the robot localization in ambiguous environments is expressed as:

$$Bel(\mathbf{x}_t, e_t) = P(\mathbf{x}_t, e_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}_a, \mathbf{m}_g) \quad (12)$$

Also, based on the independent assumptions embedded in the improved DBN, we have:

$$\begin{aligned} Bel(\mathbf{x}_t, e_t) &= P(\mathbf{x}_t, e_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}_a, \mathbf{m}_g) \\ &= \eta P(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}_g) P(\mathbf{x}_t, e_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}, \mathbf{m}_a, \mathbf{m}_g) \\ &= \eta P(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}_g) \iint P(\mathbf{x}_t, e_t, \mathbf{x}_{t-1}, e_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}, \mathbf{m}_a, \mathbf{m}_g) d\mathbf{x}_{t-1} de_{t-1} \\ &= \eta P(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}_g) \iint P(\mathbf{x}_t, e_t | \mathbf{x}_{t-1}, e_{t-1}, \mathbf{u}_t, \mathbf{m}_a) P(\mathbf{x}_{t-1}, e_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}, \mathbf{m}_a, \mathbf{m}_g) d\mathbf{x}_{t-1} de_{t-1} \\ &= \eta P(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}_g) \iint P(e_t | \mathbf{x}_t, e_{t-1}, \mathbf{m}_a) P(\mathbf{x}_t | \mathbf{x}_{t-1}, e_{t-1}, \mathbf{u}_t, \mathbf{m}_a) Bel(\mathbf{x}_{t-1}, e_{t-1}) \end{aligned} \quad (13)$$

As compared to Equation (11), we conclude that Equations (11) and (13) share the same observation model $P(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}_g)$. A different motion model $P(\mathbf{x}_t | \mathbf{x}_{t-1}, e_{t-1}, \mathbf{u}_t, \mathbf{m}_a)$ considering the cumulative ambiguity error e_{t-1} and the \mathbf{m}_a are imposed in the improved DBN. Meanwhile, Equation (13) has an additional term $P(e_t | \mathbf{x}_t, e_{t-1}, \mathbf{m}_a)$ to estimate the cumulative ambiguity error. It is noted that the multiple integrations in Equation (13) cannot be derived analytically. In our paper, the AMCL is adopted to inference the robot's pose as well as the accumulated localization error, which utilized a set of weighted particles to estimate the probability distribution. Therefore, the probability distribution $P(\mathbf{x}_t, e_t, \mathbf{x}_{t-1}, e_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}, \mathbf{m}_a, \mathbf{m}_g)$ can be reformulated according to [40] as:

$$P(\mathbf{x}_t, e_t, \mathbf{x}_{t-1}, e_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}, \mathbf{m}_a, \mathbf{m}_g) = \sum_{i=1}^{n_p} \omega_t^i \delta([\mathbf{x}_t, e_t, \mathbf{x}_{t-1}, e_{t-1}] - [\mathbf{x}_t^i, e_t^i, \mathbf{x}_{t-1}^i, e_{t-1}^i]) \quad (14)$$

where $[\mathbf{x}_t^i, e_t^i, \mathbf{x}_{t-1}^i, e_{t-1}^i]$ denotes a sample of robot state also called particle, ω_t^i stands for the i -th particle weight, δ is the Dirac delta function and n_p is the number of particles.

Substituting Equation (14) into the second part of Equation (13) yields:

$$\begin{aligned} P(\mathbf{x}_t, e_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}_a, \mathbf{m}_g) &= \iint P(\mathbf{x}_t, e_t, \mathbf{x}_{t-1}, e_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}, \mathbf{m}_a, \mathbf{m}_g) d\mathbf{x}_{t-1} de_{t-1} \\ &= \sum_{i=1}^{n_p} \omega_t^i \iint \delta([\mathbf{x}_t, e_t, \mathbf{x}_{t-1}, e_{t-1}] - [\mathbf{x}_t^i, e_t^i, \mathbf{x}_{t-1}^i, e_{t-1}^i]) d\mathbf{x}_{t-1} de_{t-1} \\ &= \sum_{i=1}^{n_p} \omega_t^i \delta([\mathbf{x}_t, e_t] - [\mathbf{x}_t^i, e_t^i]) \end{aligned} \quad (15)$$

4.3. Portal Motion Model for Improved DBN

Intuitively, the prediction step of the AMCL generates some particles in an unambiguous area which is adjacent to the ambiguous area, the update and resampling steps of the AMCL method will make the particles immediately converge to the robot's true pose when the robot leaves the ambiguous area. To accommodate that, we rewrite the motion model $P(\mathbf{x}_t | \mathbf{x}_{t-1}, e_{t-1}, \mathbf{u}_t, \mathbf{m}_a)$ as:

$$P(\mathbf{x}_t | \mathbf{x}_{t-1}, e_{t-1}, \mathbf{u}_t, \mathbf{m}_a) = (1 - \alpha_t) P_{odo}(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) + \alpha_t P_{portal}(\mathbf{x}_t | \mathbf{x}_{t-1}, e_{t-1}, \mathbf{m}_a) \quad (16)$$

where $P_{odo}(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ is the odometry motion model, α_t is a weighting factor and $P_{portal}(\mathbf{x}_t | \mathbf{x}_{t-1}, e_{t-1}, \mathbf{m}_a)$ is the portal motion model to generate the particles in an adjacent unambiguous area, which is defined as:

$$P_{portal}(\mathbf{x}_t | \mathbf{x}_{t-1}, e_{t-1}, \mathbf{m}_a) = \begin{cases} \eta_{hit} e^{-\frac{1}{2}(\mathbf{x}_t - \mu_{hit})^T \Sigma_{hit}^{-1} (\mathbf{x}_t - \mu_{hit})}, & \text{for } aae(\mathbf{p}_{x_t}) \leq t_a \\ 0, & \text{for } aae(\mathbf{p}_{x_t}) > t_a \end{cases} \quad (17)$$

where η_{hit} is a normalization coefficient, $aae(\mathbf{p}_{x_t})$ denotes the AAE of the position \mathbf{p}_{x_t} for the pose \mathbf{x}_t , t_a is a threshold and $aae(\mathbf{x}_t) \geq t_a$ implies the robot is in an ambiguous area. t_a is set to χ_1 which denote the maximum tolerable distance error. It is concluded from Equation (17) that given $\mathbf{x}_{t-1}, e_{t-1}$ and \mathbf{m}_a ,

x_t obeys a truncated Gaussian distribution. Specifically, in an ambiguous area, the probability of x_t is zero whereas it obeys a Gaussian distribution with a mean μ_{hit} and a covariance matrix Σ_{hit} in an unambiguous area.

The portal motion model describes a state transition that a robot may be immediately moved to someplace in the adjacent unambiguous area regardless of the odometer measurement. Despite the physically impossible, it can be used to obtain more reliable pose prediction in an ambiguous environment so that the accumulated localization error can be corrected immediately when the robot reaches the adjacent unambiguous area. As a complement to the odometer motion model, the portal motion model explicitly indicates the effect of cumulative localization error caused by perception aliasing on prediction robot pose.

A schematic diagram for the portal motion model is shown in Figure 6. We define $d_{range} \triangleq e_{t-1}$ as the cumulative ambiguity error at the time $t-1$, and its length is shown with a blue line. Assuming the robot pose at the time $t-1$ is x^1 which is in the ambiguous area, μ_{hit} can be generated by a simulated beam with ray casting method [41]. Practically, the simulated beam continues unless the endpoint of the beam hits an unambiguous area or an obstacle or the length of the beam reaches d_{range} . As shown in Figure 6, the red line shows the simulated beam generated from x^1 and the length of the beam is represented as d_{hit} . If the beam ends in an unambiguous area, the position and the orientation of μ_{hit} can be determined by the position and orientation of the beam, respectively. And if the beam ends because of the other two conditions, the portal motion model is not enabled. Define σ_p and σ_θ representing the adjustable standard deviation of the position and orientation, respectively. Large σ_p and σ_θ means large uncertainties of the truncated Gaussian distribution. Then, we can calculate Σ_{hit} by:

$$\Sigma_{hit} = \begin{bmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_p^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] \begin{bmatrix} 1 & 0 & 0 \\ 0 & k_{hit} & 0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]^{-1} \quad (18)$$

where \mathbf{v}_1 is a unit vector with the same orientation as x^1 , \mathbf{v}_2 is the unit vector perpendicular to \mathbf{v}_1 , $\mathbf{v}_3 = (0, 0, 1)^T$, k_{hit} is a stretch factor that used to adjust the ratio between the feature vectors of the covariance matrix Σ_{hit} given by:

$$k_{hit} = \begin{cases} k_{max}, & \text{for } \frac{d_{range}}{d_{hit}} \geq k_{max} \\ \frac{d_{range}}{d_{hit}}, & \text{for } \text{else} \end{cases} \quad (19)$$

where k_{max} is the parameter to limit the ratio between the feature vectors of the covariance matrix Σ_{hit} .

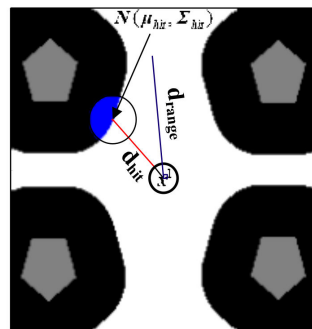


Figure 6. Schematic diagram of the portal motion model. The white area, black area, and gray area represent an ambiguous area, an unambiguous area, and area occupied by obstacles, respectively. x^1 is a pose sample. The blue transparent circle is the Gaussian distribution $N(\mu_{hit}, \Sigma_{hit})$, and the blue opaque portion expresses the truncated Gaussian distribution of Equation (17).

Figure 7a–c depict the changing process of Σ_{hit} . It can be seen that as the pose of the robot changes from x^1 to x^3 , d_{hit} gradually decreases ($d_{hit}^1 > d_{hit}^2 > d_{hit}^3$). According to Equation (18), the uncertainties of the truncated Gaussian distribution in the direction perpendicular to the robot's orientation gradually increase due to the decreasing of d_{hit} , as represented by the blue oval area. We adopt the KLD-sampling method such that few and relatively concentrated particles will be generated by the portal motion model when the robot is in an ambiguous area and is far away from an unambiguous area. Moreover, as the distance between the robot and the unambiguous area getting shorter, more dispersed particles will be generated by the portal motion model due to large uncertainties of Σ_{hit} . With an adaptive Σ_{hit} , we can reasonably add uncertainties for the portal motion model and improves the reliability of localization.

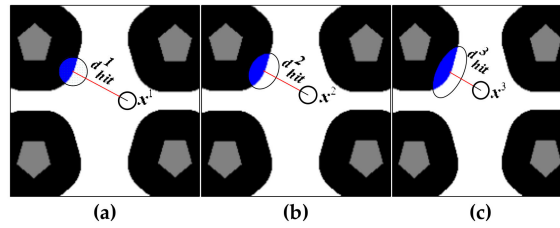


Figure 7. Changing process of Σ_{hit} . The uncertainty of $N(\mu_{hit}, \Sigma_{hit})$ is increasing as a robot moves in an ambiguous area and more particles will be generated by the portal motion model.

4.4. AGM-AMCL Implementation

For implementation, $P(e_t|x_t, e_{t-1}, \mathbf{m}_a)$ is designed using a deterministic function:

$$e_t = \begin{cases} e_{t-1} + \frac{aae(\mathbf{p}_{x_t})}{\rho \cdot e_{t-1} + 1} & \text{for } aae(\mathbf{p}_{x_t}) > t_a \\ 0 & \text{for } aae(\mathbf{p}_{x_t}) \leq t_a \end{cases} \quad (20)$$

where $\rho \geq 0$ and is a parameter to control the changing rate of e_t . By Equation (20), we show that e_t in an ambiguous area and an unambiguous area is derived as $e_t = e_{t-1} + aae(\mathbf{p}_{x_t}) / (\rho \cdot e_{t-1} + 1)$ and $e_t = 0$, respectively.

Using the length of the beam d_{hit} , we schedule the α_t adaptively as:

$$\alpha_t = \begin{cases} 0 & \text{for } (d_{hit} = d_{range}) \vee (d_{hit} = 0) \\ 1 - \frac{d_{hit}}{d_{range}} & \text{for } 0 < d_{hit} < d_{range} \end{cases} \quad (21)$$

As demonstrated in Equation (21), if the beam ends due to the length restriction or the zero-length of d_{hit} which means the beam end because of obstacle or the pose x_t in an unambiguous area, α_t is assigned to zero, otherwise, it is regulated in real-time where $0 < d_{hit} < d_{range}$ indicates that the robot is in an ambiguous area and an unambiguous area is found in the scope of d_{range} .

A detailed overview of the AGM-AMCL algorithm can be found in Algorithm 1. The output particle set is initialized as an empty set in Step 1. Step 2 declares a variable C_t to store the ray casting result to avoid repetitive computation and is initialized from Step 3 to Step 5. Step 7 to Step 23 show the prediction step of our localization method which is derived from Equation (16). μ_{hit} and d_{hit} are obtained by the ray casting method at step 10, α_t is calculated according to Equation (21) at step 15. At step 16, c is a binary variable sampled from a Bernoulli distribution with parameter α_t . $c = 1$ means that the prediction pose will be sampled from the portal motion model while $c = 0$ means the prediction pose will be sampled from the odometry motion model. At step 18, Σ_{hit} is computed according to Equation (18). Through step 19 and 20, the portal model is used to sample a prediction pose. According to Equation (20), the cumulative ambiguity error e_t is updated at step 24. In order to improve the efficiency of the particles, the KLD sampling method presented in [38] is adopted at step 28.

Algorithm 1 AGM-AMCL

Input: $S_{t-1} = \left\{ \left\langle \mathbf{x}_{t-1}^{[i]}, e_{t-1}^{[i]}, \omega_{t-1}^{[i]} \right\rangle_{i=1}^{N_{t-1}} \right\}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{m}_g, \mathbf{m}_a$

Output: $S_t = \left\{ \left\langle \mathbf{x}_t^{[i]}, e_t^{[i]}, \omega_t^{[i]} \right\rangle_{i=1}^{N_t} \right\}$

- 1: $S_t \leftarrow \emptyset, N_x \leftarrow 0, n \leftarrow 0, \beta \leftarrow 0$ // Initialization
- 2: $C_t = \left\{ \left\langle \mathbf{x}_{hit}^{[i]}, d_{hit}^{[i]}, \text{tag}^{[i]} \right\rangle_{i=1}^{N_{t-1}} \right\}$ //For storing RAYCAST result
- 3: for $i \leftarrow 1, \dots, N_{t-1}$ do //Initialize C_t
- 4: $\text{tag}^{[i]} \leftarrow 0$
- 5: end for
- 6: do
- 7: $j \leftarrow \text{RESAMPLING}(S_{t-1})$ //Resampling according to particle weight
- 8: $d_{range} = e_{t-1}^{[j]}$
- 9: if $\text{tag}^{[j]} == 0$ then
- 10: $(\boldsymbol{\mu}_{hit}, d_{hit}) \leftarrow \text{RAYCAST}(\mathbf{x}_{t-1}^{[j]}, d_{range}, \mathbf{m}_a)$
- 11: $C_t[j] \leftarrow (\mathbf{x}_{hit}, d_{hit}, 1)$
- 12: else
- 13: $(\boldsymbol{\mu}_{hit}, d_{hit}) \leftarrow C_t[j]$
- 14: end if
- 15: $\alpha \leftarrow \text{COMPUTEALPHA}(d_{hit}, d_{range})$
- 16: sample c according to Bernoulli_Distribution(α)
- 17: if $c == 1$
- 18: $\Sigma_{hit} \leftarrow \text{COMPUTESIGMAHITT}(d_{range}, d_{hit}, \mathbf{x}_{t-1}^{[j]})$
- 19: Create $P_{portal}(\mathbf{x}_t | \mathbf{x}_{t-1}^{[j]}, e_{t-1}^{[j]}, \mathbf{m}_a)$ according to $N(\boldsymbol{\mu}_{hit}, \Sigma_{hit})$
- 20: Sample $\mathbf{x}_t^{[n]}$ according to $P_{portal}(\mathbf{x}_t | \mathbf{x}_{t-1}^{[j]}, e_{t-1}^{[j]}, \mathbf{m}_a)$
- 21: else
- 22: Sample $\mathbf{x}_t^{[n]}$ according to $P_{odo}(\mathbf{x}_t | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t)$
- 23: end if
- 24: $e_t^{[n]} \leftarrow \text{PREDICTAMBIGUITYERROR}(\mathbf{x}_t^{[n]}, e_{t-1}^{[j]}, \mathbf{m}_a)$
- 25: $\omega_t^{[n]} \leftarrow P(\mathbf{z}_t | \mathbf{x}_t^{[n]}, \mathbf{m}_a)$ //Compute weight according to observation
- 26: $\beta \leftarrow \beta + \omega_t$
- 27: $S_t \leftarrow S_t \cup \left\{ \left\langle \mathbf{x}_t^{[n]}, e_t^{[n]}, \omega_t \right\rangle \right\}$
- 28: $N_x \leftarrow \text{ADPTINGSAMPLESIZE}(\mathbf{x}_t^{[n]})$
- 29: $n \leftarrow n + 1$
- 30: while $n < N_x$ and $n < N_{\min}$
- 31: for $i \leftarrow 1, \dots, n$ do //Normalize particle weight
- 32: $\omega_t^{[i]} \leftarrow \omega_t^{[i]} / \beta$
- 33: end for
- 34: return S_t

5. Experiments and Results

In this section, we will verify the effectiveness of the AGM and the performance of the AGM-AMCL algorithm severally. Note that, if not particularly indicated, AAE represents the AAE of a position in this section.

5.1. Platform and Environments

The experimental platform and environments are shown in Figure 8, respectively. The used mobile robot named Turtlebot2 is equipped with a Hokuyo UTM-30LX LIDAR and an LPMS-RS232AL2 IMU, which serve as the external and internal sensor, separately. As depicted in Figure 8a,b, three different ambiguous environments with long corridors, open square and tangled thicket are considered for the real-world experiments.

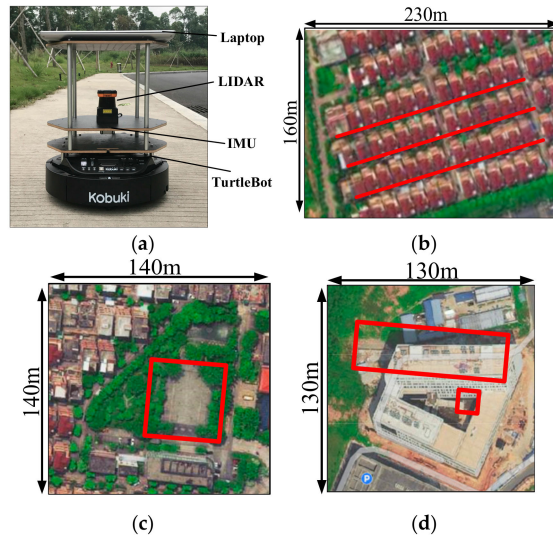


Figure 8. Experimental platform and real-world environments: (a) Turtlebot2 for data capturing; (b) scene A: long corridors; (c) scene B: empty space; (d) scene C: tangled thicket. The red lines in scene A show three parallel long corridors, and each corridor is surrounded by two parallel walls without prominent features. The red square in scene B shows an open square. The red square in scene C show area with tangled thicket.

5.2. AGM of Artificial Environments

To verify the effectiveness of the AGM, we draw several grid maps, in which the localizability of different position can be easily distinguished manually. In our research, the log-likelihood is used for observation likelihood computation. And is set to 0.5 which is a very small value for log-likelihood and is useful to compensate for some numerical calculation error. χ_1 and χ_2 were selected as 0.25 m and 5° respectively in this experiment, therefore the maximum AAE value is 0.5 m. As demonstrated in Figure 9, the top row and bottom row present three artificial occupancy grid maps and their corresponding AGM, respectively. Low AAE value (blue) of a cell means small localization error while high AAE value (red) means large localization error. When building these AGMs, the range error of the laser scan simulator was set to 0 m to simulate the laser scan data. The map shown in Figure 9(a1) is a long corridor. It can be seen from its AGM that the AAE value in both ends of the corridor is small, and the ε AAE value in the middle of the corridor is high, which is according with human experience. The map shown in Figure 9(b1), is a circle. When a robot is at the center of the circle, it will obtain the same sensor readings regardless of the orientation of the robot. Therefore, as shown in Figure 9(b2), the AAE value at the center of the circle is very high. An open space environment is shown in Figure 9(c1). We can see from its AGM that the farther a position is away from the obstacle, the higher the AAE value. For those cells where the AAE value is higher than one-half of the maximum AAE value (In this AGM experiment for artificial environments, the maximum AAE value is 0.5 m), the AAE value shown in Figure 9(c2) is significantly higher than the AAE value shown in Figure 9(a2,b2). This is because in the open space of Figure 9(c1), a robot actually cannot recognize anything and it cannot localization itself. But in Figure 9(a1,b1), the robot is only confusing in some

directions and the sensor reading obtained from those positions can be used to reduce some of the localization uncertainties.

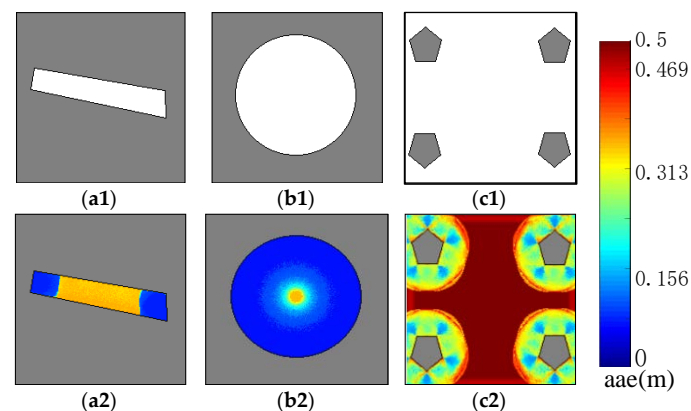


Figure 9. AGMs of virtual ambiguous environments: (a1–c1) are the occupancy grid maps of three artificial environments. (a2–c2) are the corresponding AGMs. AGMs show that the middle area of (a1), the center area of (b1) and the open area of (c1) get high AAE value because of their ambiguity.

Figure 10 shows the performance of the AGM of an unambiguous environment under the condition that the simulated laser scan data have different range error. Figure 10a shows the occupancy grid map. Figure 10b,c are the corresponding AGM of Figure 10a when the range error of the laser scan simulator is set to 0 m and 0.5 m, respectively. By comparing Figure 10b with Figure 10c, we can clearly see that even in the same environment, large range error will lead to large AAE value. Although the environment shown in Figure 10a does not seem to be an ambiguous environment, experiment results show that large range error of the laser scan data may still lead to perception aliasing, which demonstrates the importance of sensor noise for localizability evaluation.

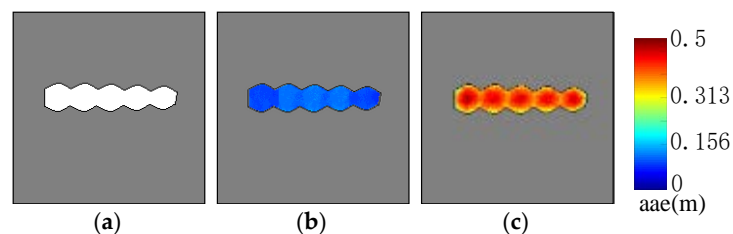


Figure 10. AGMs with different range error for virtual unambiguous environments: (a) Occupancy grid map of an unambiguous environment. (b) AGM with zero distance error. (c) AGM with 0.5 m distance error. The figure clearly shows that large distance error leads to large AAE value.

5.3. AGM of Real Environments

The SLAM method proposed in [42] was used to build the occupancy grid maps of scene A, B and C which are shown in Figure 11(a1–c1), respectively. In fact, it is difficult to build the occupancy grid maps of these three environments because of their ambiguity. However, with the help of those vehicles parked on the side of the road, localization can be achieved during mapping. Moreover, loop closure and offline optimization adopted by the SLAM method enable us to build consistent occupancy grid maps. After mapping, those vehicles are manually removed from the occupancy grid maps due to the fact that the position of those parked vehicles had been changed when we carried out our localization experiment. The resolution of our occupancy grid maps was set as 0.2 m. In addition, χ_1 and χ_2 are selected as 0.5 m and 10° , respectively, in our experiments. The true noise parameters of the Hokuyo UTM-30LX 2D Laser range finder are adopted to generate more realistic laser scan data for computing AAE.

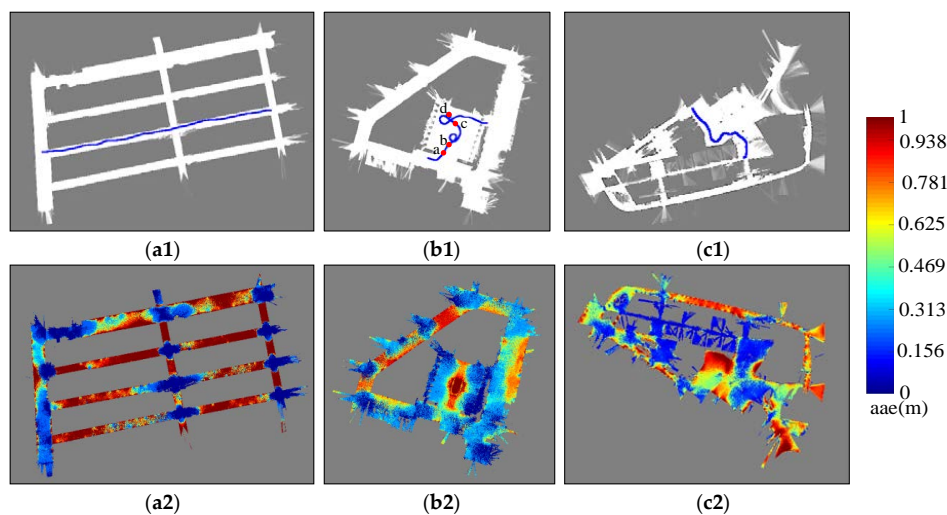


Figure 11. AGMs of real-world environments: (a1–c1) are the occupancy grid maps of scene A, B and C. (a2–c2) are the corresponding AGMs. Blue curves show the ground truth for localization simulation. Those positions in the middle of a long corridor, in the center of an open square, near a straight-line wall and tangled thicket get high AAE value.

It can be seen from the AGM, shown in Figure 11(a2–c2) that those positions which are in the middle of a long corridor, in the center of the open square, near a straight-line wall and tangled thicket get high AAE value. Those positions near a corner get low AAE. The results are in line with human experience.

5.4. Localization Simulation

The aim of the simulation experiments is to comprehensively evaluate the performance of our proposed AGM-AMCL method and compare it with the AMCL method and the augmented MCL method proposed in [37]. In fact, the augmented MCL is very similar to the method proposed in [33]. The difference is that extra random particles are added in [37] while sensor resetting method was adopted to generate extra particles in [33], which is not suitable in large scale and ambiguous environments. The trajectories (Figure 11) which are selected to pass through an ambiguous area are used for all simulation experiments. Laser scan data used for localization simulation can be generated at each pose on the simulation trajectories with the help of the occupancy grid maps. t_a is set to 0.5 which is the maximum tolerable distance error. k_{max} is selected to 2 and ρ is set to 0.5. These three parameters were used in all of our experiments. In addition, the localization methods are initialized around the true position in our simulation experiments only to verify the pose tracking performance.

Figure 12 presents the evolution process of particles during localization simulation in scene B. The color of a particle visualizes the particle weight just like the color used in Figure 10 except that the range represented by the color is 0 to 1.

In the beginning, those particles (Figure 12(a1)) generated by AMCL can follow the true pose of the robot but as the robot entering an ambiguous area, the particles (Figure 12(b1)) became divergent and there were fewer and fewer particles (Figure 12(c1)) near the true pose of the robot during the robot moving through an ambiguous area. Even the robot entered an unambiguous area, the particles (Figure 12(d1)) generated by AMCL could not correct the accumulated localization error and converged to a wrong pose. For augmented MCL, extra random particles Figure 12(b2–d2) are added when the average likelihood of particles was decreased. However, those extra random particles led the method to converge to a wrong place in large scale and ambiguous environment. This result shows that it is essential to generate particles in proper locations, otherwise wrong convergence could occur. On the other hand, as the robot entered the ambiguous area, our localization method generate extra particles (Figure 12(c3)) in the adjacent unambiguous area when the accumulated localization error is large

enough. Once the robot reached the unambiguous area, those extra particles (Figure 12(d3)) help the robot to correct its accumulated localization error immediately.

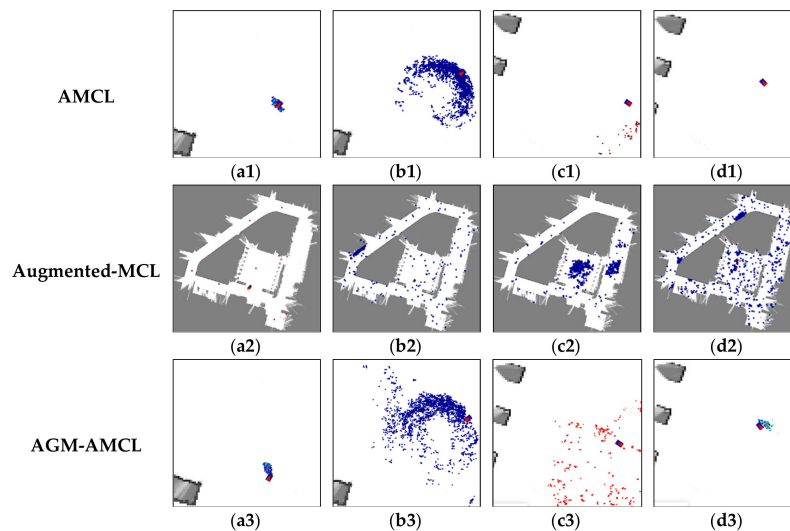


Figure 12. Evolution of particles using AMCL (top), Augmented-MCL (middle) and AGM-AMCL (bottom) in scene B. The letters a, b, c and d denote four locations shown in Figure 11(b1) with red dots. The blue square with a red triangle inside represents the true pose and the colored points denote particles. With some particles in the nearby unambiguous area, our method can converge to the true pose after a robot moving through an ambiguous area.

Figures 13–15 present the localization simulation results of three localization methods using the trajectory shown in Figure 11. Distance error and orientation error are shown in (a) and (b). Number of particles and the running time are shown in (c) and (d). In terms of the distance error, our method has better performance than the other two methods. The distance error of AMCL is divergent when the robot moved through the ambiguous areas while the distance error of augmented MCL fluctuate greatly because of the randomly added particles. This is because the orientation can be easily determined in the long corridor while it will cause large distance error. The computational complexity of those three methods is related to the number of particles. However, the particle number of our method and AMCL is stochastic, which rely on the sensor measurement data during robot moving. For this reason, it is hard to evaluate the efficiency theoretically. From the results of three different environments, the efficiency of our method can be demonstrated in three aspects. First, the running time of our method is similar to AMCL in unambiguous environments by using the AGM. Second, because of the right convergence, our method is more efficiency after the robot moving through an ambiguous environment. Third, our method is slower than AMCL when the robot is in an ambiguous area and near an unambiguous area, but it is still enough for real-time localization.

In order to evaluate the localization reliability of our method, we consider the last localization result after a robot traveling through a trajectory. If the distance error of the last localization result is smaller than 0.5 m and the angular error of the last localization result is smaller than 10° , we assume this localization experiment is successful, otherwise, this localization experiment fails. Then the reliability of our localization method can be verified through the localization success rate of repeated localization experiments on the same trajectory. Various degrees of odometer errors were adopted in our reliability experiment. The true odometry data is computed according to:

$$u_t^{true} = \ominus x_{t-1}^{true} \oplus x_t^{true} \quad (22)$$

where \ominus and \oplus are the pose compounding operator [43], x_{t-1}^{true} and x_t^{true} are the true pose of a robot at time $t - 1$ and time t , respectively. x_{t-1}^{true} and x_t^{true} can be obtained from the simulated trajectory.

Taking the true odometer data as the mean and setting a covariance matrix, a Gaussian distribution can be obtained. The odometer data with errors can be obtained by sampling from such Gaussian distribution. In our experiments, the abovementioned covariance matrix is set to a diagonal matrix with one parameter:

$$\Sigma_{sample} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & 0.2 * m \end{bmatrix} \quad (23)$$

Then we define the odometry error rate η as:

$$\zeta = \frac{m}{\text{distance}(x_t^{true}, x_{t-1}^{true})} \quad (24)$$

By changing ζ , the odometer data with different degrees of error can be obtained. In our reliability experiments, eleven different value of ζ varying from 0 to 1 were selected to evaluate the localization reliability under different degree of odometry error. In addition, given ζ and a trajectory, we carried out one hundred repetitive experiments to get the localization success rate. The results of the reliability experiments with different odometry errors are shown in Figure 16.

As shown in these results, the localization success rate of AGM-AMCL is significantly higher than that of AMCL and augmented-MCL under different odometry error in three scenes. The minimum success rate of AMCL, augmented-MCL and AGM-AMCL are 6%, 0% and 79%, respectively. Moreover, the localization success rate of AMCL and augmented-MCL decrease rapidly with the increase of odometry error in three scenes while the localization success rate of AGM-AMCL is always 100% in scene A and slightly decreases in scene B and C. The results of the reliability experiments show that our localization method is more reliable when locating a robot in ambiguous environments.

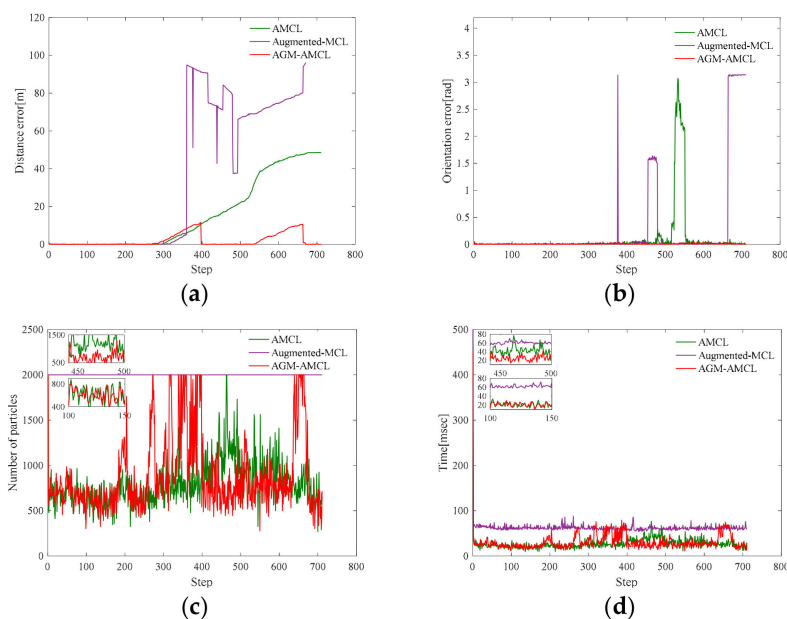


Figure 13. Localization results in scene A: (a) distance error; (b) orientation error; (c) number of particles; (d) running time. Note that the orientation errors of AMCL are small in most times. This is because the orientation can be easily determined in the long corridor while it will cause large distance error. The running time of our method is similar to AMCL in unambiguous areas (the zoomed result from step 100 to 150). Our method is more efficient than AMCL after the robot moving through an ambiguous area (the zoomed result from step 450 to 500). Our method needs more particles when the robot is in an ambiguous area and near an unambiguous area (step 300 to 400), therefore it is less efficient than AMCL. Because of the fixed particle number, augmented-MCL takes more time than the other two methods in most time.

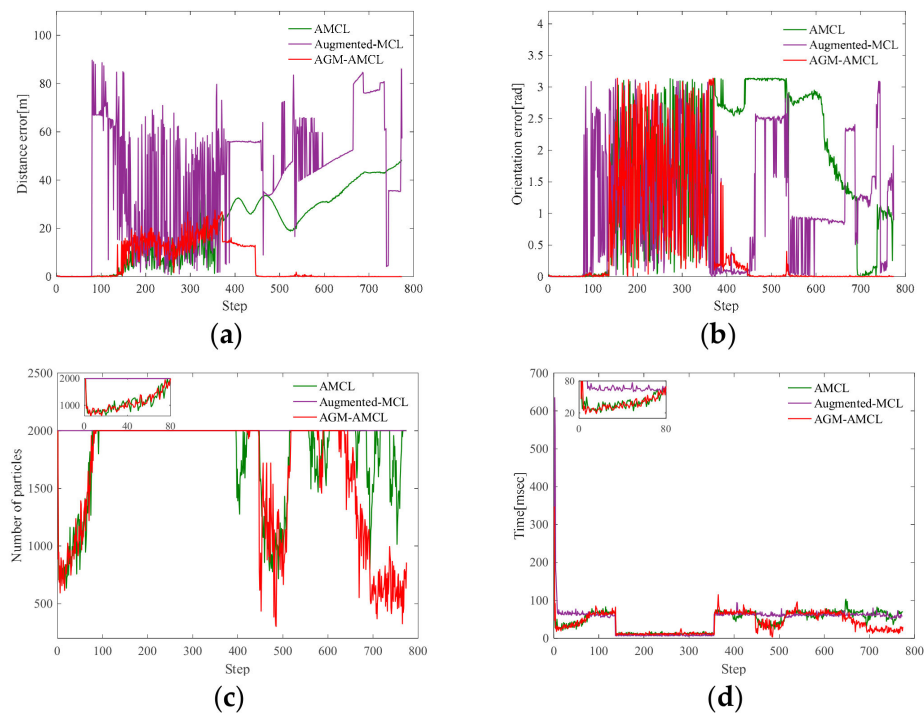


Figure 14. Localization results in scene B: (a) distance error; (b) orientation error; (c) number of particles; (d) running time. From step 700 to 800, our method is more efficient because of the right convergence. The zoomed results from step 0 to 80 show that our localization method has similar running time with AMCL in unambiguous area.

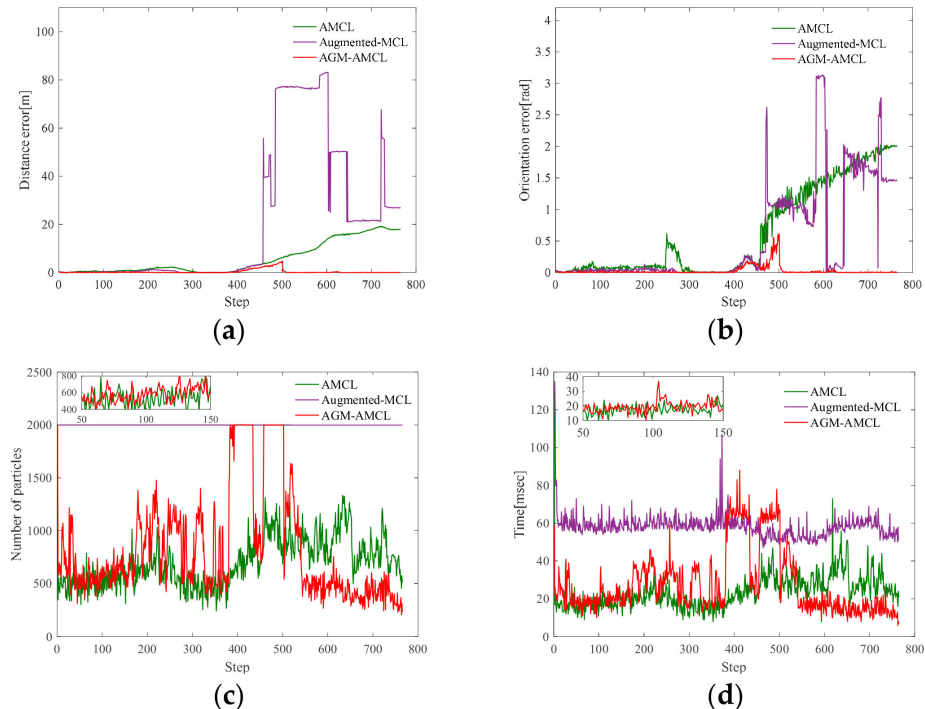


Figure 15. Localization results in scene C: (a) distance error; (b) orientation error; (c) number of particles; (d) running time. The zoomed results from step 50 to 150 show that our localization method has similar running time with AMCL. Our method takes less time from step 600 to 800 while more time is need from step 400 to 550.

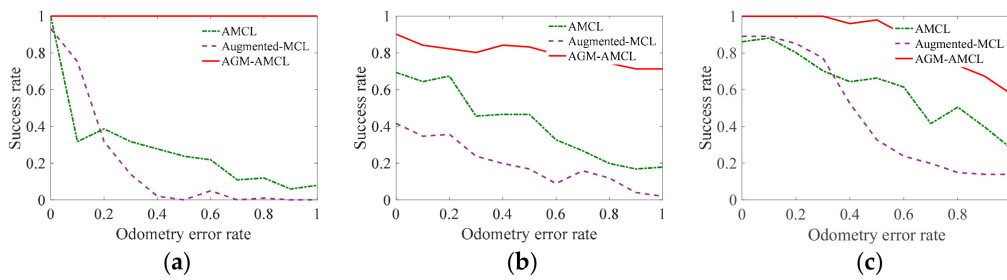


Figure 16. The success rate of three localization methods with different odometry errors: (a) Results in Scene A. (b) Results in Scene B. (c) Results in Scene C. The results show that our method has a considerable success rate even in the case when the odometry error is high in ambiguous environments.

5.5. Localization Using Real-World Data

Based on the real-world laser scan data and the odometry data captured by our robot in scene A, B and C, the localization results of three localization methods are obtained, as shown in Figure 17. As we can see from these figures, AMCL and augmented-MCL failed to track the robot pose due to the fact that the registered laser scan data obtained by AMCL and augmented-MCL are inconsistent with the maps. On the contrary, the registered laser scan data obtained by our method is consistent with the maps and the resulting trajectories are closed successfully which are in line with the actual robot trajectories. The inconsistent laser scan data registration obtained by AGM-AMCL is caused by dynamic obstacles and localization error. From the trajectories generated by AGM-AMCL, it can be seen that some discontinuous points appeared on the neighboring locations of an ambiguous area and an unambiguous area. This is because the accumulated localization error was corrected by our localization method after the robot moving through an ambiguous area, which demonstrates that our localization method is more reliable in real-world ambiguous environments.

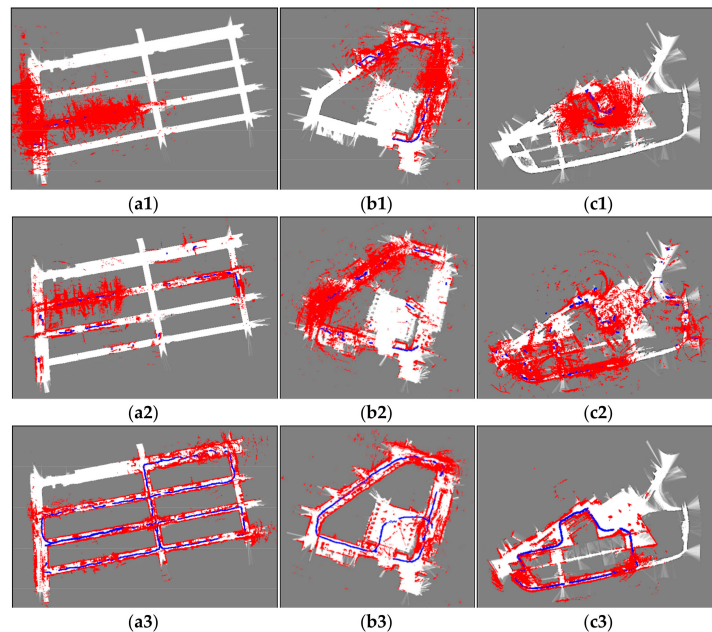


Figure 17. Localization results using real-world data. (a1–c1) show the localization result using AMCL. (a2–c2) show the localization result using augmented-MCL. (a3–c3) show the localization result using our method. Blue curves show the robot trajectories obtained by localization methods. Red point clouds represent the registered laser scan data using the trajectories. Based on the trajectories obtained by our method, laser scan data can be successfully registered in three ambiguous environments. The discontinuity of the trajectories obtained by our method is caused by the portal motion model.

6. Conclusions

In this paper, an AGM-based AMCL method to achieve fast and reliable localization of a robot in ambiguous environments was presented. Our method was derived from an improved DBN which introduces two nodes to represent the accumulated ambiguity error and the AGM separately. The AGM models the ambiguity of an environment, and it can be used to predict the accumulated localization error during pose reasoning. Moreover, utilizing the AGM, a portal motion model was implemented to obtain more reliable pose prediction in an ambiguous area. We evaluated the effectiveness of the AGM and the AGM-AMCL method through simulation and real-world experiments. The results demonstrated that the proposed AGM is in line with human experience and our localization method can achieve fast and reliable localization in ambiguous environments, as compared with conventional methods.

In the future, we will improve the AGM to model the ambiguity in dynamic environments so as to address the localization problem in an environment where the ambiguity is not only caused by symmetrical or featureless areas but also by dynamic objects.

Author Contributions: Conceptualization, G.L.; Data curation, J.M., X.Z., L.J. and C.L.; Formal analysis, G.L.; Funding acquisition, Y.H.; Methodology, G.L.; Project administration, Y.H.; Resources, Y.H.; Software, G.L.; Supervision, Y.H.; Validation, J.M., Y.X. and L.J.; Visualization, X.Z. and C.L.; Writing—original draft, G.L.; Writing—review & editing, J.M. and Y.X.

Funding: This work was supported in part by Dongguan Innovative Research Team Program under Grant 201536000100031, in part by Science and Technology Planning Project of Guangdong Province under Grant 2017B090913001, in part by National Natural Science Foundation of China under Grant 51805190, in part by China Postdoctoral Science Foundation under Grant 2019M650179 and in part by Guangdong major science and technology project under Grant 2019B090919003.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lajoie, P.; Hu, S.; Beltrame, G.; Carlone, L. Modeling Perceptual Aliasing in SLAM via Discrete-Continuous Graphical Models. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1232–1239. [[CrossRef](#)]
2. Tang, J.; Chen, Y.; Jaakkola, A. NAVIS-An UGV Indoor Positioning System Using Laser Scan Matching for Large-Area Real-Time Applications. *Sensors* **2014**, *14*, 11805–11824. [[CrossRef](#)] [[PubMed](#)]
3. Pinto, A.M.; Moreira, A.P.; Costa, P.G. A localization method based on map-matching and particle swarm optimization. *J. Intell. Robot. Syst.* **2015**, *77*, 313–326. [[CrossRef](#)]
4. Maffei, R.; Jorge, V.A.M.; Rey, V.F.; Kolberg, M.; Prestes, E. Fast Monte Carlo Localization using spatial density information. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 6352–6358.
5. Saarinen, J.; Andreasson, H.; Stoyanov, T.; Lilienthal, A.J. Normal distributions transform Monte-Carlo localization (NDT-MCL). In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 382–389.
6. Guan, R.P.; Ristic, B.; Wang, L.; Palmer, J.L. KLD sampling with Gmapping proposal for Monte Carlo localization of mobile robots. *Inf. Fusion* **2019**, *49*, 79–88. [[CrossRef](#)]
7. Hanten, R.; Buck, S.; Otte, S.; Zell, A. Vector-AMCL: Vector based adaptive monte carlo localization for indoor maps. In Proceedings of the International Conference on Intelligent Autonomous Systems, Shanghai, China, 3–7 July 2016; pp. 403–416.
8. Xu, S.; Chou, W.; Dong, H. A Robust Indoor Localization System Integrating Visual Localization Aided by CNN-Based Image Retrieval with Monte Carlo Localization. *Sensors* **2019**, *19*, 249. [[CrossRef](#)] [[PubMed](#)]
9. Vasiljević, G.; Miklič, D.; Draganjac, I.; Kovačić, Z.; Lista, P. High-accuracy vehicle localization for autonomous warehousing. *Robot. Comput. Integr. Manuf.* **2015**, *42*, 1–16. [[CrossRef](#)]
10. Li, G.; Meng, J.; Xie, Y.L.; Zhang, X.L.; Jiang, L.Q.; Huang, Y. An Improved Observation Model for Monte-Carlo Localization Integrated with Reliable Reflector Prediction. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Hong Kong, China, 8–12 July 2019.

11. Meyer-Delius, D.; Beinhofer, M.; Kleiner, A.; Burgard, W. Using artificial landmarks to reduce the ambiguity in the environment of a mobile robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 5173–5178.
12. Cheikhrouhou, O.; Bhatti, G.; Alroobaea, R. A Hybrid DV-Hop Algorithm Using RSSI for Localization in Large-Scale Wireless Sensor Networks. *Sensors* **2018**, *18*, 1469. [[CrossRef](#)] [[PubMed](#)]
13. Zhang, L.; Zapata, R.; Lepinay, P. Self-adaptive Monte Carlo localization for mobile robots using range finders. *Robotica* **2012**, *30*, 229–244. [[CrossRef](#)]
14. Oh, S.M.; Tariq, S.; Walker, B.N.; Dellaert, F. Map-based priors for localization. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 28 September–2 October 2004; pp. 2179–2184.
15. Censi, A. On achievable accuracy for range-finder localization. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 4170–4175.
16. Censi, A. On achievable accuracy for pose tracking. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 1–7.
17. Liu, Z.; Chen, W.; Wang, Y.; Wang, J. Localizability estimation for mobile robots based on probabilistic grid map and its applications to localization. In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Hamburg, Germany, 13–15 September 2012; pp. 46–51.
18. Irani, B.; Wang, J.; Chen, W. A Localizability Constraint-Based Path Planning Method for Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 2593–2604. [[CrossRef](#)]
19. Pilia, V.; Gupta, K. Localization aware sampling and connection strategies for incremental motion planning under uncertainty. *Auton. Robot.* **2017**, *41*, 111–132. [[CrossRef](#)]
20. Zhen, W.; Zeng, S.; Soberer, S. Robust localization and localizability estimation with a rotating laser scanner. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 6240–6245.
21. Akai, N.; Morales, L.Y.; Takeuchi, E.; Yoshihara, Y.; Ninomiya, Y. Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching. In Proceedings of the IEEE Intelligent Vehicles Symposium, Los Angeles, CA, USA, 11–14 June 2017; pp. 6240–6245.
22. Javanmardi, E.; Javanmardi, M.; Gu, Y.; Kamijo, S. Factors to Evaluate Capability of Map for Vehicle Localization. *IEEE Access* **2018**, *6*, 49850–49867. [[CrossRef](#)]
23. Beinhofer, M.; Kretschmar, H.; Burgard, W. Deploying artificial landmarks to foster data association in simultaneous localization and mapping. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 5235–5240.
24. Chen, Y.; Francisco, J.; Trappe, W.; Martin, R.P. A practical approach to landmark deployment for indoor localization. In Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks, Reston, VA, USA, 28–29 September 2006; pp. 365–373.
25. Mi, J.; Takahashi, Y. Design of an HF-Band RFID System with Multiple Readers and Passive Tags for Indoor Mobile Robot Self-Localization. *Sensors* **2016**, *16*, 1200. [[CrossRef](#)]
26. Liu, F.; Zhong, D. GSOS-ELM: An RFID-Based Indoor Localization System Using GSO Method and Semi-Supervised Online Sequential ELM. *Sensors* **2018**, *18*, 1995. [[CrossRef](#)]
27. Zhang, Q.; Wang, P.; Chen, Z. An improved particle filter for mobile robot localization based on particle swarm optimization. *Expert Syst. Appl.* **2019**, *135*, 181–193. [[CrossRef](#)]
28. Wolcott, R.W.; Eustice, R.M. Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving. *Int. J. Robot. Res.* **2017**, *36*, 292–319. [[CrossRef](#)]
29. Blanco, J.L.; González, J.; Fernández-Madrigal, J.-A. A robust multi-hypothesis approach to matching occupancy grid maps. *Robotica* **2013**, *31*, 687–701. [[CrossRef](#)]
30. Martín, F.; Carballeira, J.; Moreno, L.; Garrido, S.; González, P. Using the Jensen-Shannon, Density Power, and Itakura-Saito Divergences to Implement an Evolutionary-Based Global Localization Filter for Mobile Robots. *IEEE Access* **2017**, *5*, 13922–13940. [[CrossRef](#)]
31. Park, S.; Roh, K.S. Coarse-to-Fine Localization for a Mobile Robot Based on Place Learning With a 2-D Range Scan. *IEEE Trans. Robot.* **2016**, *32*, 528–544. [[CrossRef](#)]
32. Rottmann, A.; Mozos, O.M.; Stachniss, C.; Burgard, W. Place classification of indoor environments with mobile robots using boosting. In Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, PA, USA, 9–13 July 2005; pp. 1306–1311.

33. Gutmann, J.S.; Fox, D. An Experimental Comparison of Localization Methods Continued. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; pp. 454–459.
34. Thrun, S.; Fox, D.; Burgard, W.; Dellaert, F. Robust Monte Carlo localization for mobile robots. *Artif. Intell.* **2001**, *128*, 99–141. [[CrossRef](#)]
35. Wang, J.; Wang, P.; Chen, Z. A novel qualitative motion model based probabilistic indoor global localization method. *Inf. Sci.* **2018**, *429*, 284–295. [[CrossRef](#)]
36. Pfaff, P.; Plagemann, C.; Burgard, W. Gaussian mixture models for probabilistic localization. In Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 467–472.
37. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic robotics*; MIT Press: Cambridge, MA, USA, 2005; pp. 153–169.
38. Fox, D. Adapting the Sample Size in Particle Filters through KLD-Sampling. *Int. J. Robot. Res.* **2003**, *22*, 985–1003. [[CrossRef](#)]
39. Biswas, J.; Veloso, M.M. Episodic non-Markov localization. *Robot. Autom. Syst.* **2017**, *87*, 162–176. [[CrossRef](#)]
40. Särkkä, S. *Bayesian Filtering and Smoothing*; Cambridge University Press: Cambridge, UK, 2013; pp. 120–125.
41. Even, J.; Furrer, J.; Morales, Y.; Ishi, C.T.; Hagita, N. Probabilistic 3-D Mapping of Sound-Emitting Structures Based on Acoustic Ray Casting. *IEEE Trans. Robot.* **2017**, *33*, 333–345. [[CrossRef](#)]
42. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
43. Corke, P. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*; Springer: Cham, Switzerland, 2011; pp. 32–40.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).