



Enhanced resampling scheme for Monte Carlo localization

Suat Karakaya¹

Received: 15 November 2023 / Accepted: 26 February 2024 / Published online: 25 March 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

The indoor positioning problem is a critical research domain essential for real-time control of mobile robots. Within this field, Monte Carlo-based solutions have been devised, leveraging the processing of diverse sensor data to address numerous challenges in local and global positioning. This study focuses on resampling strategies within the conventional Monte Carlo framework, which directly impact positioning performance. From this perspective, in contrast to the conventional method of employing weight thresholding and full particle scattering when the robot becomes disoriented, this study proposes an alternative approach. It advocates for a localized space resampling strategy, adaptive noise injection guided by likelihood, and the incorporation of beam rejection modifications to address dynamic (unmapped) obstacles effectively. The real-time experimental results, conducted with varying particle counts, demonstrate that the proposed scheme effectively manages the presence of unmapped obstacles while employing fewer particles than the standard Monte Carlo implementation.

Keywords Mobile robotics · Localization · Likelihood

1 Introduction

The pursuit of accurate and efficient indoor localization for mobile robots is a fundamental challenge in the field of robotics. It involves the intricate task of determining a robot's precise position and orientation in real-time, relative to a predefined map of the environment and incoming sensor data. At its core, indoor localization represents the task of compensating for sensor noise, requiring the mobile robot to deduce its state from inherently noisy and indirectly observable information [1]. Over the years, the localization problem has garnered substantial attention within the realm of robotics, establishing itself as a pivotal and primary concern in the context of autonomous guided vehicle (AGV) navigation and broader mobile robotic applications [2]. This multifaceted problem can be categorized into two distinct scenarios: the initial knowledge of the robot's pose narrows uncertainty to a local scale, transforming the pose estimation problem into a localized position tracking challenge. When the robot's initial pose remains unknown, it engenders global localization, giving rise to a more extensive realm of uncertainty. In a contrasting scenario, when

an autonomous robot finds itself relocated to an arbitrary position during operation, the problem is further complicated, leading to the so-called “kidnapped robot problem” [3]. This latter scenario poses formidable challenges beyond those of global localization. Within the extensive landscape of indoor localization techniques, a diverse array of methods has been proposed and explored, encompassing approaches such as the Extended Kalman Filter (EKF) [4], grid-based algorithms [5], multi-hypothesis tracking [6], and the Monte Carlo Localization (MCL) [7] methodology, among others. Notably, MCL stands as a prominent subset of techniques designed to effectively address the challenge of uncertainty. It leverages a probabilistic framework to compute the instantaneous uncertainty of a robot, proving versatile for both local and global pose estimation problems. Nevertheless, conventional MCL is not without its limitations. As it represents the posterior probability distribution by a collection of weighted samples via particle filtering, the number of samples required for global localization can be substantial, elevating the algorithm's computational complexity and resource demands [8]. Furthermore, the resampling process, employed to counteract localization failures or unexpected sensor noise, poses its own set of challenges. Augmented MCL attempts to alleviate such issues by introducing random samples throughout the map, but this stochastic approach can inadvertently discard poses in close proximity to the true robot's position,

✉ Suat Karakaya
suatronik@gmail.com

¹ Kocaeli University, 41001 Izmit, Kocaeli, Turkey

thereby impairing real-time performance. MCL has found widespread adoption in mobile robotics and autonomous vehicle navigation. It plays a crucial role in enabling robots to estimate their positions within complex and dynamic environments. Researchers have applied MCL techniques to ground robots, aerial drones, and underwater vehicles, showcasing its adaptability across various robotic platforms [9]. Indoor localization is a particularly challenging domain due to the absence of GPS signals and the presence of intricate, cluttered environments. MCL has proven to be a reliable solution for indoor localization, offering the ability to fuse data from diverse sensors, such as range finders, cameras, and inertial measurement units (IMUs). This adaptability has made MCL a popular choice for robots navigating indoor spaces [10, 11]. MCL has also found applications in augmented reality, where it aids in aligning virtual objects with the real world, enhancing user experiences. Additionally, it is frequently employed in Simultaneous Localization and Mapping (SLAM) algorithms, which are essential for building maps of unknown environments while simultaneously localizing the robot within them [12–14]. MCL's flexibility transcends environmental boundaries. While it is extensively used for outdoor applications like autonomous ground vehicles and drones navigating in urban or rural settings, it also excels in indoor environments where GPS signals are often unavailable. In indoor scenarios, MCL's ability to fuse data from a combination of sensors, including LiDAR, IMUs, and visual cameras, has made it indispensable for robots operating in cluttered spaces [15]. In multi-robot systems, accurate localization information is critical for efficient coordination and collaboration among robots. MCL has been employed to synchronize the movements and actions of multiple robots in applications ranging from exploration missions to surveillance and industrial automation [16, 17]. It is important to note that MCL is not confined to range sensors alone; researchers have extended its application to include vision sensors, utilizing techniques such as Gist descriptors and Principal Component Analysis (PCA) to compute likelihoods based on visual information [18].

This paper introduces new methodologies aimed at reducing uncertainty in the global indoor localization problem. To cope with the drawback of scattering particles across the entire map resulting from unmapped obstacles, adaptive noise injection and model-based prior focusing, and beam rejection approaches have been proposed. In Sect. 2, the theoretical framework of the standard Monte Carlo approach is presented. Section 3 expounds upon the proposed enhancements integrated into conventional Monte Carlo localization. In this section, basic information about an external indoor positioning system [19] used as a reference for absolute positioning is presented. The Image Processing-based Indoor Positioning System (IPS) has been employed as a reference

position source for both the standard Monte Carlo Localization (MCL) and modified MCL approaches. The performance of both particle-based IPS methods is evaluated using the Image Processing IPS in identical indoor scenarios. The results obtained in the laboratory environment and in-depth discussions are presented in Sect. 4.

2 Theoretical framework of MCL

This chapter examines the details of Monte Carlo localization (MCL). Through the use of random sampling and statistical inference, MCL offers a sophisticated resolution to the formidable task of achieving precise localization within a pre-established map. In order to understand the conventional Monte Carlo localization approach in detail, it becomes necessary to attempt a mathematical explanation. This effort will elucidate the formalized complexities and computational procedures that underlie this widespread method of robotic localization. The core concept of Monte Carlo Localization is to estimate the posterior probability distribution of the robot's state X_t given the history of sensor measurements $Z_{1:t}$ and control inputs $U_{1:t}$, as well as the map M . This is expressed using Bayesian inference:

$$P(X_t|Z_{1:t}, U_{1:t}, M) \propto P(Z_t|X_t, M) \int P(X_t|X_{t-1}, U_t, M) P(X_{t-1}|Z_{1:t-1}, M) dX_{t-1} \quad (1)$$

The sensor model $P(Z_t|X_t, M)$ represents the likelihood of observing sensor measurements Z_t given the current state X_t and the map M . It incorporates both the noise in the sensor measurements and the correspondence between map features and observed landmarks. A complex sensor model might involve Gaussian distributions, such as:

$$P(Z_t|X_t, M) = \prod_i \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(z_i - h_i(X_t))^2}{2\sigma_i^2}} \quad (2)$$

where

- z_i is the observed measurement,
- $h_i(X_t)$ is the predicted measurement based on the current state X_t and the map feature i .
- σ_i is the sensor noise for measurement i .

The motion model $P(X_t|X_{t-1}, U_t, M)$ describes how the state X_t evolves from the previous state X_{t-1} under the influence of control input U_t and the map M . This can involve complex motion equations, such as:

$$X_t = g(U_t, X_{t-1}, M) + \varepsilon_t \quad (3)$$

where

- $g(\dots)$ is a complex function representing the robot’s motion dynamics.
- ε_t is the motion noise.

To approximate the posterior distribution, a set of particles is used, $X_t^{(i)}$, sampled from the motion model and weighted according to the sensor model:

$$\omega_t^{(i)} = \frac{P(Z_t | X_t^{(i)}, M)}{\sum_j (Z_t | X_t^{(j)}, M)} \tag{4}$$

After weighting, particles are resampled with replacement according to their weights to generate a new set of particles. This step ensures that particles representing regions with higher likelihoods are maintained, while less likely hypotheses are pruned.

The final state estimate is typically computed as a weighted sum of the resampled particles:

$$X_t^* = \sum_i \omega_t^{(i)} X_t^{(i)} \tag{5}$$

MCL strategy, with its theoretical foundation elaborated in (1) to (5), is depicted as pseudo-code in Algorithm 1.

Algorithm 1: Pseudo code of Monte Carlo localization

```

Initialize particles randomly in the map
Set initial belief about the robot's pose
While not converged:
    Move the robot based on control input
    Update particle poses based on motion model
    Add noise to particle poses
    Measure the environment and calculate particle weights
    Resample particles according to their weights
    Estimate the robot's pose from the particles
    Check for termination condition (e.g., convergence or a maximum number of iterations)
End
    
```

3 Proposed improvements on resampling strategy

This section delves into an examination of the variables influencing the algorithm’s robustness, along with the introduction of strategies for addressing these factors, including noise injection, particle confinement, and beam rejection. The contents of this section outline the study’s advancements as an improved Monte Carlo localization scheme and how

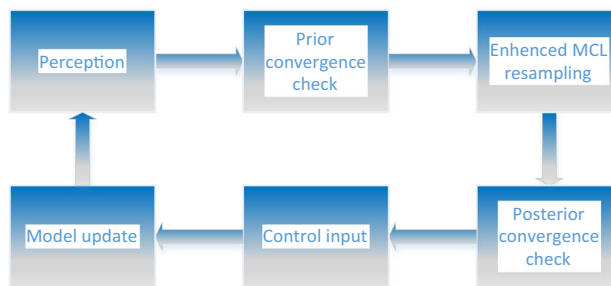


Fig. 1 Operating loop of the enhanced MCL resampling

it extends beyond conventional methods, both theoretically and visually. Basic operating loop of the proposed scheme is given in Fig. 1.

In the context of particle-based localization methodologies, a critical step involves the resampling of the prior particle set. This resampling process is characterized by the random selection of a subset of particles from the existing distribution. The primary objective of this operation is to refine the particle ensemble in accordance with the evolving belief about the estimated position or state of a target entity. The resampling process ensures that the particle set is updated to better represent the posterior probability distribution. It is widely recognized as an essential component in various localization and tracking systems across different domains, contributing significantly to the enhancement of estimation accuracy and reliability. By allowing for the ran-

dom selection of particles from the current distribution, the resampling procedure conforms to the principles of statistical inference and Bayesian filtering, where the representation of posterior beliefs is paramount in achieving accurate localization results. With the enhancement of convergence, the particle overlap narrows the diversity within the algorithm’s solution space. This phenomenon can result in the algorithm becoming ensnared by localized solutions, thereby giving rise to persistent steady-state errors. The noise injection approach is introduced in the domain of MCL, where the standard deviation of injected noise during resampling is

Fig. 2 Likelihood-based sampling of particles

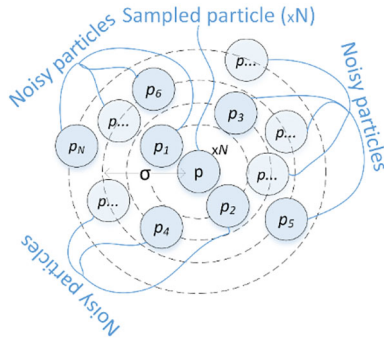
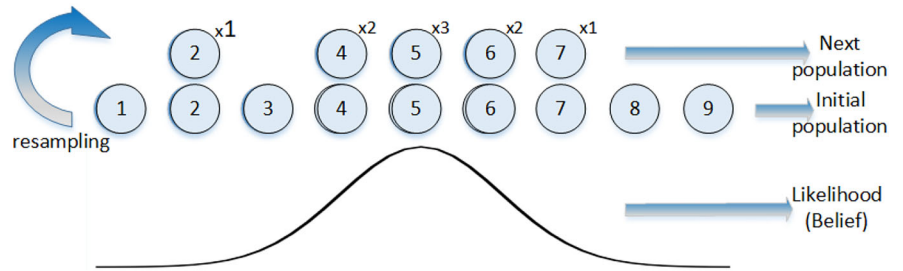


Fig. 3 Diversified concentric particles with noise injection

adaptively modulated based on the convergence levels of particle weights, aiming to improve the precision and efficiency of the localization process. The standard deviation added to increase diversity is adjusted according to the convergence level to provide a wide scan for low weights and a narrower scan for high weights.

$$\sigma_t = A\left(\left\{\omega_t^{(i)}\right\}\right) \tag{6}$$

where σ_t is the standard deviation of the noise for time step t , and $\left\{\omega_t^{(i)}\right\}$ is the set of particle weights.

Then adding this noise when resampling the particles:

$$X_t^{(i)} \sim P\left(X_t|X_{t-1}^{(i)}, U_t, M\right) + \mathcal{N}\left(0, \sigma_t^2\right) \tag{7}$$

where $\mathcal{N}\left(0, \sigma_t^2\right)$ represents Gaussian noise with mean 0 and standard deviation σ_t .

The adaptive noise term ensures that particles with lower weights receive higher noise, while particles with higher weights receive lower noise. This leads that the noise injected into the particles is adjusted based on the convergence level as indicated by the weights.

The depiction of the sampling process grounded in likelihood-based beliefs is presented in Fig. 2.

In the sequential progression following the procedure elucidated in Figs. 2 and 3 provides a visual representation of the introduction of noise via a dedicated injection process into the concentric particles, situated within the framework

of diversification. The theoretical foundation of this process is explicated in Eq. 7.

When the robot’s position achieves a significant level of convergence, reductions in particle weights may transpire due to environmental dynamics. For convergence levels below a critical threshold, standard MCL resamples particles across the entire unoccupied space. While this approach is effective for reaching the global optimum in theory, practical constraints arise from the robot’s kinematics. Consequently, it is unnecessary to scan such an extensive span. Thus, an approximate sub-space for the next-step position can be delineated within the framework of known control inputs, such as velocity, and the robot’s kinematic model. Within the framework of Monte Carlo Localization, a strategy is introduced whereby the dispersion of particles is constrained to a circular domain, with its center situated at the last verifiable robot position. This novel approach is designed to enhance localization precision by confining particle exploration within a spatially informed region of interest.

$$X_t^{(i)} = X_t^{(i)} + \text{Clamp}\left(X_t^{(i)} - X_t^*, r_m\right) \tag{8}$$

Here, “Clamp” is a function that restricts the position of each particle to stay within the circular region with radius r_m . If a particle goes beyond this circle, its pose is clamped to the boundary of the circle. This modification ensures that particles with low weights are scattered within the defined circular region around the latest trusted pose of the robot, preventing them from drifting too far from potentially correct estimates and allowing for exploration while maintaining relatively higher degree of localization accuracy. The depiction of this process is illustrated in Fig. 4.

In cases where the robot reaches a high level of convergence in localization estimation, the sudden decrease in particle weights attributed to the presence of an unmapped obstacle in its path reduces the robustness of Monte Carlo Localization in dynamic environments. In this context, "beam rejection" strategy has been proposed to solve this problem. The dispersion of particles into a pre-defined region of the entire map is considered undesirable. As a result, this strategy emphasizes the consideration of initially mapped obstacles that are integral components of the static map. The reflection

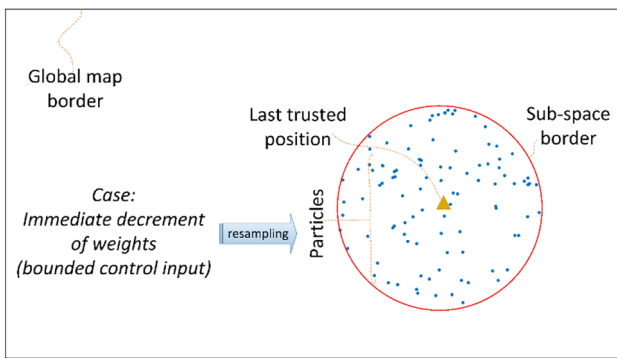


Fig. 4 Resampling within clamped sub-space

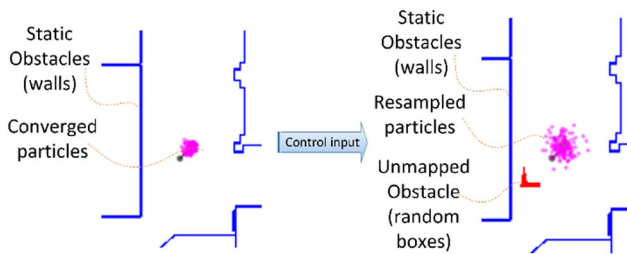


Fig. 5 Effect of unmapped obstacle on filter convergence

of laser beams off unmapped obstacles yields significantly divergent weight outcomes. Consequently, an abrupt decline in particle weights may be attributed to the presence of these uncharted obstacles. Excluding the information derived from laser beams intercepted by unmapped obstacles during the likelihood calculation process serves to maintain high levels of convergence. Assuming that the dynamic ones of these obstacles are removed from the robot’s field of vision within a certain period, the robot is ensured to maintain its converged position during this time. This results in the overlooking of the impact of these obstacles on positioning performance, even though they are perceived by the robot. The exploitation of residual data obtained from the real sensor ray casting model, in tandem with observations derived from the last reliable position on the static map, serves as an effective approach for elucidating the rays that experience reflection from unmapped obstacles.

The impact of an unmapped obstacle encountered at time $t + 1$, after the application of a control input at time t , on the convergence behavior of the filtering process is depicted in Fig. 5 within the context of a representative scenario.

The weight values (belief) of the particles at time t and $t + 1$ are given in Figs. 6 and 7, respectively. It can be observed that the initial belief decreases because the rays poorly match the unmapped obstacle, leading to a misaligned convergence pattern among particles.

Simultaneous plots of real-time sensor measurements with particle-based observation data extracted from the static

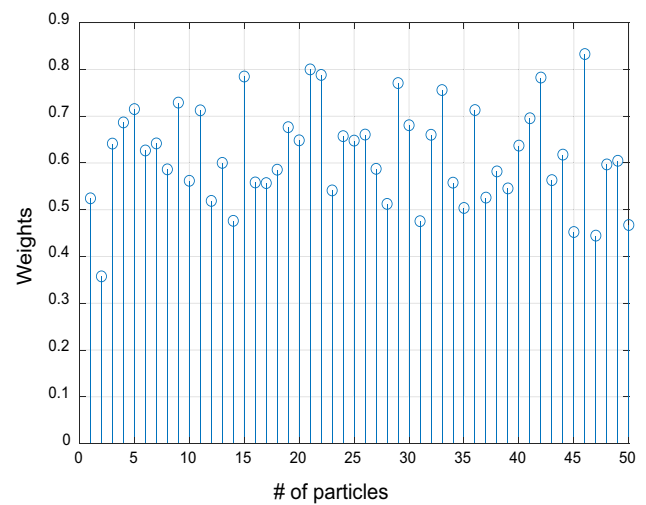


Fig. 6 Particle weight distribution at time t

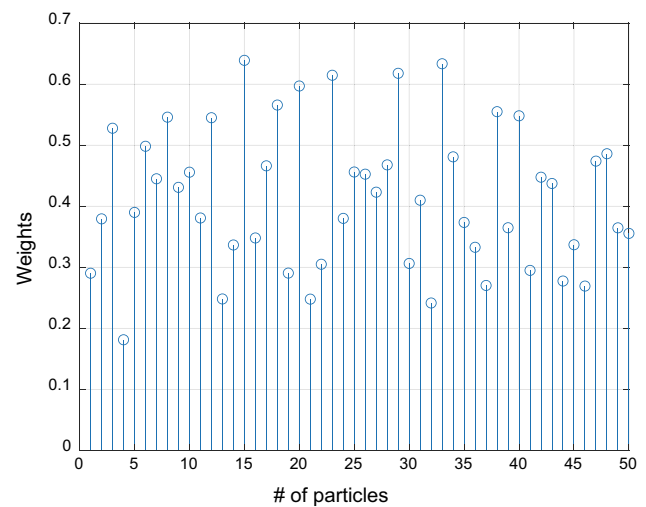


Fig. 7 Particle weight distribution at time $t + 1$

map obtained at times t and $t + 1$ are shown in Figs. 8 and 9, respectively. Poor-matched observation results resulting from the dispersion of particles can be observed from these figures. When total likelihood (production of weights) decreases significantly, a thresholding approach is employed, wherein the maximum residual value from the last reliable state is used to threshold the difference between the best particle and the actual measurement. Observation data above the threshold residual are not included in the likelihood calculation. In Fig. 10, the residual graph is presented, and it allows the identification of measurement indices that exceed the threshold at moment $t + 1$. The regions that are evident in the residual analysis are marked as red solid. This determination aids in selecting the lidar measurements to be included in the likelihood calculation, thus preventing particle dispersion, and ensuring a reliable pose estimation. The Results

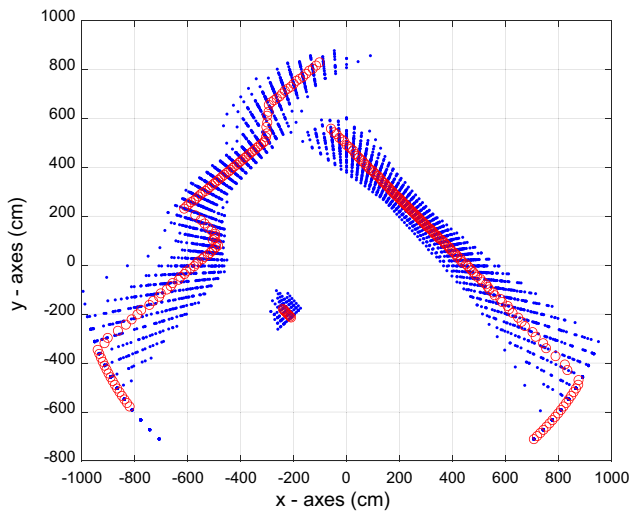


Fig. 8 Observations (‘.’) and real measurement (‘o’) at time t

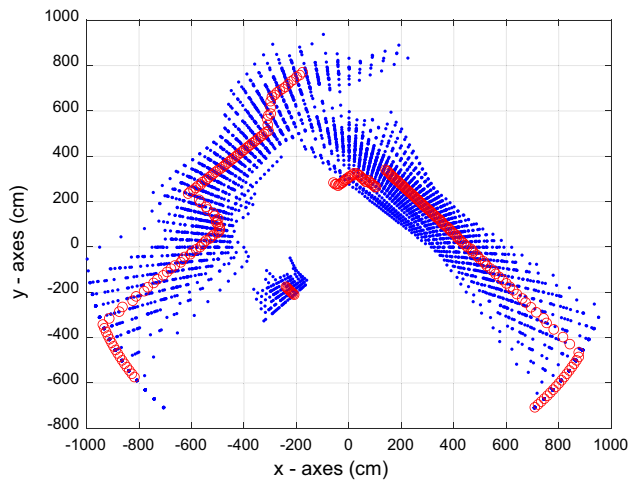


Fig. 9 Observations (‘.’) and real measurement (‘o’) at time $t + 1$

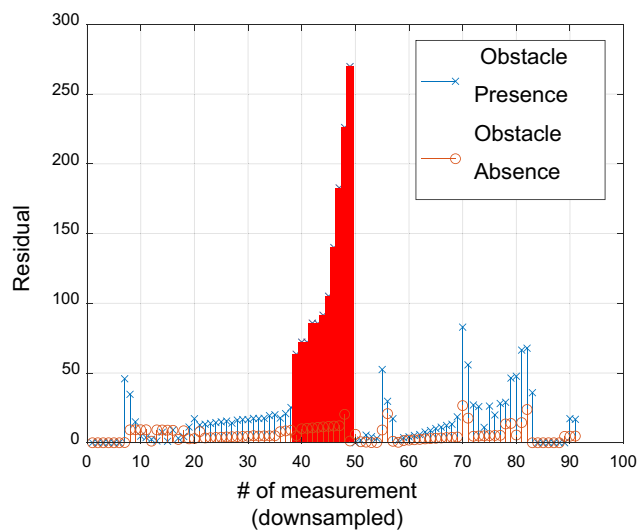


Fig. 10 Residual Analysis for Lidar Measurement Selection

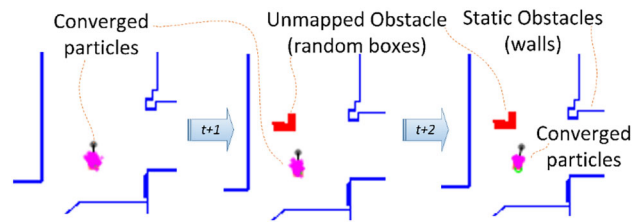


Fig. 11 Enhanced filter convergence with beam rejection scheme

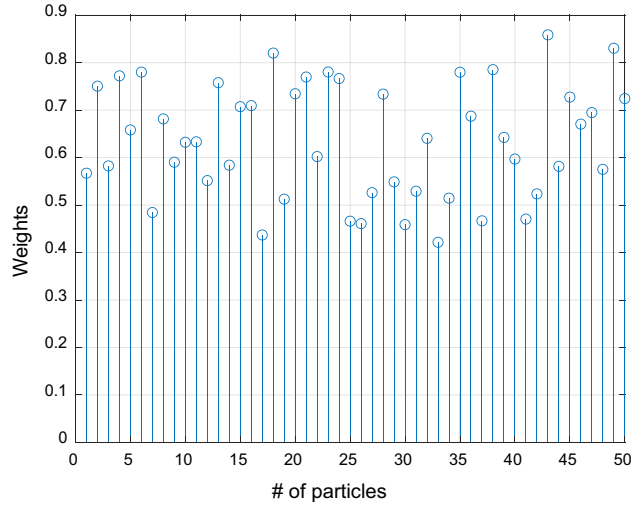


Fig. 12 Particle weight distribution at time t

Section will present experimental findings demonstrating the enhanced location accuracy achieved with this approach in the presence of unmapped obstacles. At this stage, the method for obtaining the indices of rejected beams is exclusively demonstrated.

The filter outputs obtained in a scenario where the beam rejection approach is applied are presented in Figs. 11, 12, 13 and 14, respectively. Figure 11 illustrates that the presence of unmapped obstacles does not disrupt the convergence status of the filter. The implementation of the proposed rejection scheme enables the filter to maintain its convergence status throughout subsequent time steps. Through an analysis of the weight distributions presented in Figs. 12, 13 and 14, it is evident that the prevention of undesirable weight losses and the maintenance of positioning accuracy have been achieved.

Figure 15 provides a flowchart depicting the specific stages where detailed modifications are introduced into the standard Monte Carlo.

4 Results

The indoor positioning system (IPS) given in [19] is employed to compare the position and orientation outputs, serving as a reference positioner. This system exhibits an

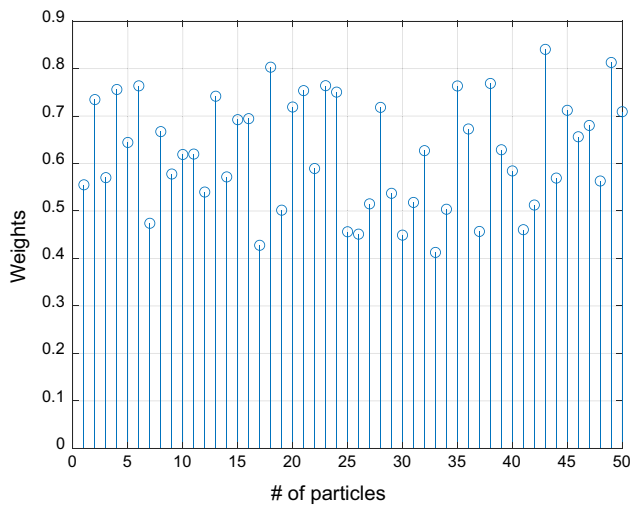


Fig. 13 Particle weight distribution at time $t + 1$

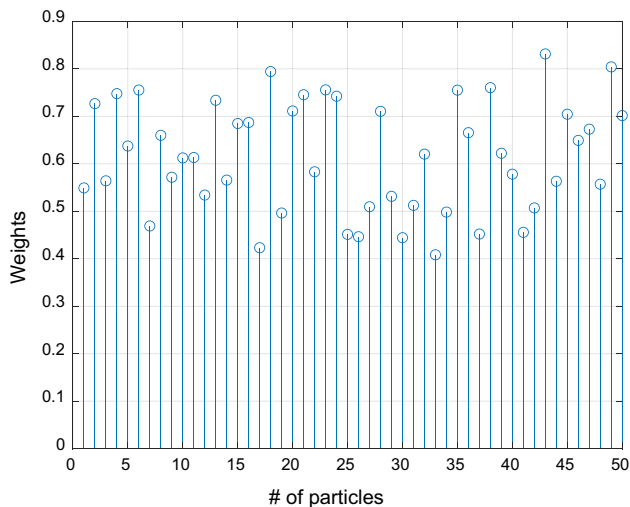


Fig. 14 Particle weight distribution at time $t + 2$

average position error of 2.3 cm and an orientation error of 0.3 degrees. The image processing-based system was deployed within the experimental corridor and data were recorded for the purpose of comparative analysis with the particle-based positioning system. The convergence ratio of the particle filter is determined based on the reference IPS. In the experimental configuration, a wheeled mobile robot (WMR) operating under a differential drive kinematic model was remotely controlled to traverse an arbitrary trajectory within the laboratory corridor. Passive IR reflectors placed on the ceiling of the corridor, and the robot equipped with LIDAR and IPS receiver are shown in Fig. 16.

The static map delineating the corridor walls was generated using SolidWorks drawing, with each pixel on the map corresponding to a 5 cm square in the physical environment.

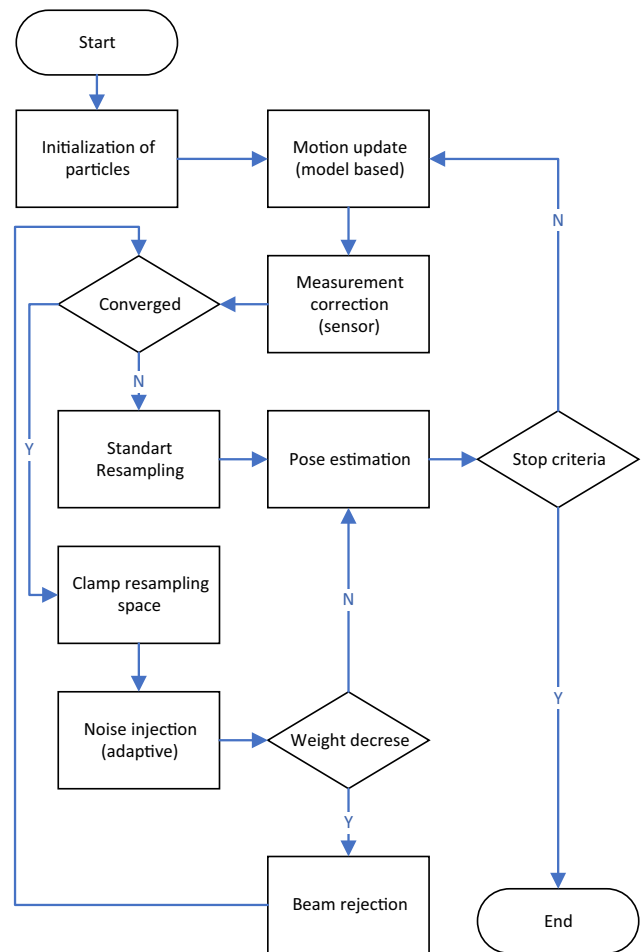


Fig. 15 Flowchart of Modified Monte Carlo localization

The actual position of the WMR was continuously monitored using the IPS throughout the experiments. Under these specified conditions, the proposed Monte Carlo Localization (MCL) algorithm was executed with various distinct particle sets, consisting of 750, 1500, 5000 and 10,000 particles, respectively.

The experimental work was conducted with the underlying assumption that there was no prior belief regarding the initial position of the wheeled mobile robot (WMR). In the scenarios where unmapped obstacles were encountered, a comparative analysis was conducted between the conventional MCL and the proposed scheme in terms of weight distribution and positioning error. Due to the absence of prior knowledge regarding the initial pose of the robot, the algorithm necessitates an initial operation over arbitrary cycles to elevate the average weights to an acceptable level. This threshold is subject to adjustment depending on the number of particles. Consequently, an investigation was conducted to determine the reliable initial average weights for particle sets comprising both 750 and 1500 particles. The initial average weight threshold is applied during the initialization step of



Fig. 16 IPS pattern and the remote controlled WMR

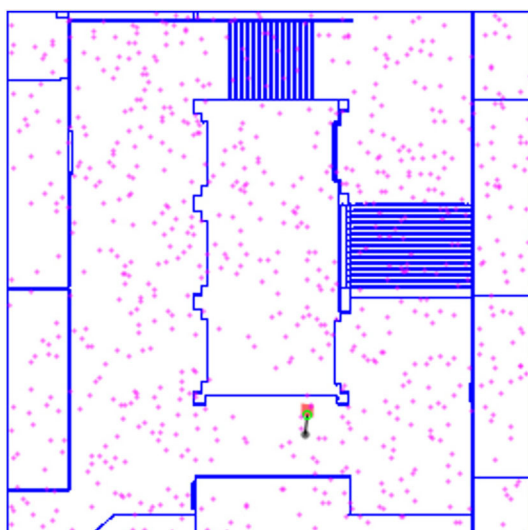


Fig. 17 Initial distribution of the particles (#750)

experimental tests as a criterion for determining immediate convergence of the filter.

The initial distribution of 750 particles is depicted in Fig. 17. To maintain a consistent scale for visualizing x – y pose errors alongside average weights, the weights were amplified by a coefficient of 3000. When examining the pose error in the X – Y dimensions alongside the corresponding weights, it becomes evident that the filter converges to the actual pose at an average weight threshold of approximately 0.58. This threshold corresponds to the 10th iteration of the algorithm. This value can be observed from the chart given in Fig. 18. If this threshold is not applied, MCL pose estimation exhibits a significant initial pose error. This phenomenon is illustrated in Fig. 19.

The visual representation in Fig. 19 includes black solid dots depicting MCL pose estimations, cyan solid dots representing IPS pose data, and a magenta point cloud enveloping the WMR icon, which corresponds to the particles. Hence, the Monte Carlo Localization (MCL) estimation cannot be

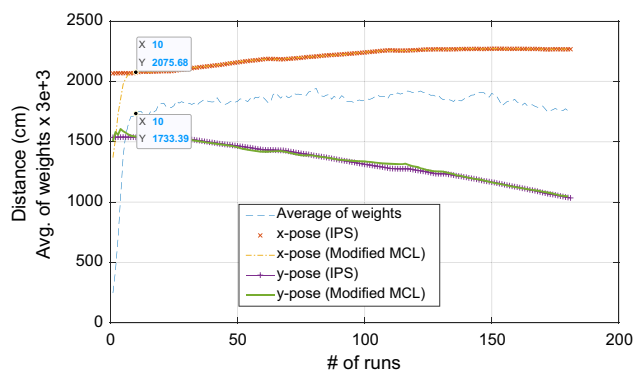


Fig. 18 Amplified average weights for visualization of x – y pose error (#750)

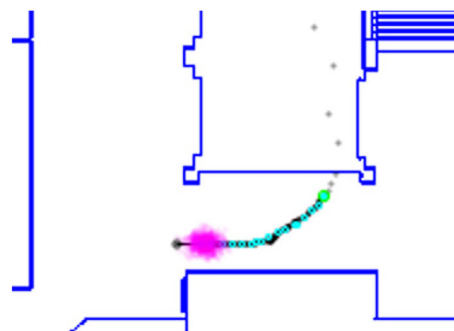


Fig. 19 High initial pose error (without weight thresholding #750)

deemed trustworthy until a specific number of initial iterations have been executed. Demonstrating this phenomenon, the pose estimation of the proposed MCL algorithm converges with IPS data, as illustrated in Fig. 20.

An equivalent experimental investigation was carried out with a particle count of 1500. The initial particle distribution is depicted in Fig. 21.

In this scenario, the threshold average weight is approximately 0.6 and this threshold corresponds to the 20th iteration of the algorithm. The index at which the x – y pose error is initially minimized can be observed in Fig. 22.

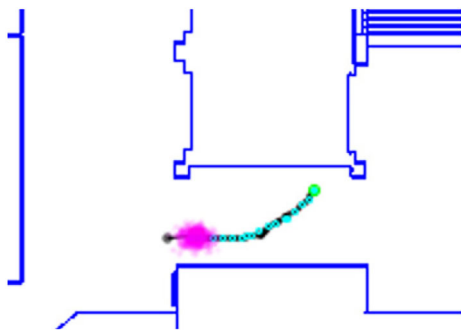


Fig. 20 Minimized initial pose error (with weight thresholding #750)

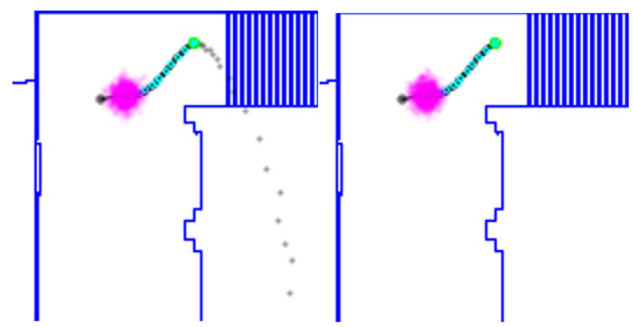


Fig. 23 Before (left) and after (right) the weight thresholding (#1500)

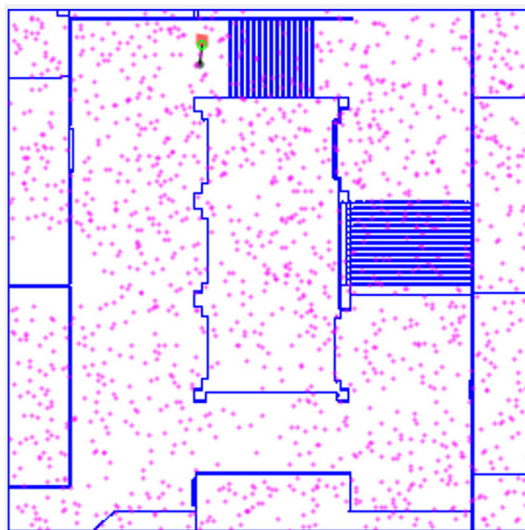


Fig. 21 Initial distribution of the particles (#1500)

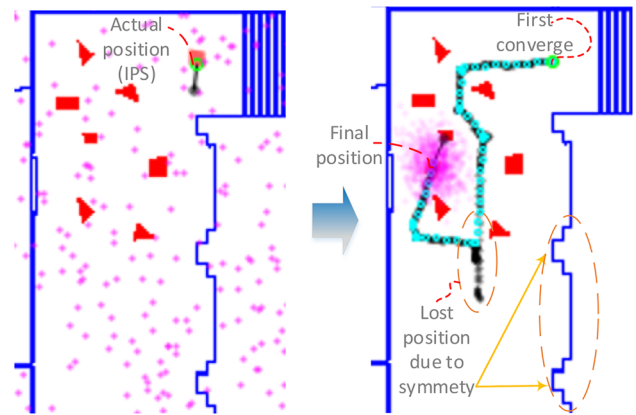


Fig. 24 Standard MCL results

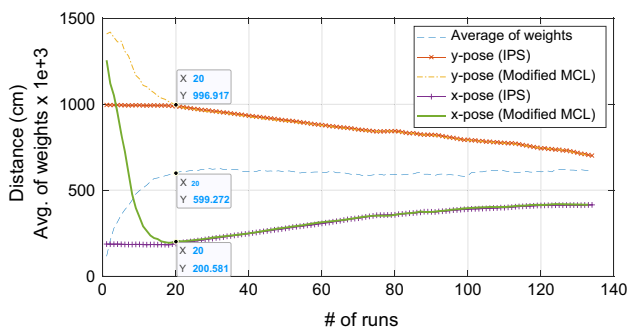


Fig. 22 Amplified average weights for visualization of x - y pose error (#1500)

The impact of the initial weight thresholding for 1500 particles is illustrated in Fig. 23.

To evaluate the efficacy of the modified MCL scheme, an unmapped obstacle configuration was established within the indoor corridor of the Laboratory. The boxes were placed in arbitrary positions, obstructing the robot’s direct line of sight to the static walls. Hence, this setup allows us to examine the

filter’s behavior when particle weights experience a reduction. In this setup, both the standard MCL and the proposed modified MCL schemes were executed to observe particle resampling behaviors in the context of weight reduction and map symmetry.

Figure 24 presents the output of the standard MCL implementation. In this execution, the actual pose received from the external IPS is represented by cyan-colored circles. The black solid icons depict MCL pose estimations at their respective time steps; while, the magenta point cloud illustrates the particle distribution. The robot, represented by a simple icon, corresponds to the WMR (Wheeled Mobile Robot). Unmapped boxes are indicated by red objects, and the static walls are delineated by blue contours.

Initially, the Wheeled Mobile Robot (WMR) was directed remotely to execute a zero-radius turn, a maneuver undertaken to facilitate the initial convergence of the filter. This controlled motion served as a deliberate action to bring the filter into a state of alignment and readiness for subsequent operations. Following that, the robot was remotely guided to navigate through the areas containing unmapped objects. Therefore, the robot confronted dynamic patterns that diverged from its prior knowledge, precipitating a reduction in the cumulative particle weight as it progressed through

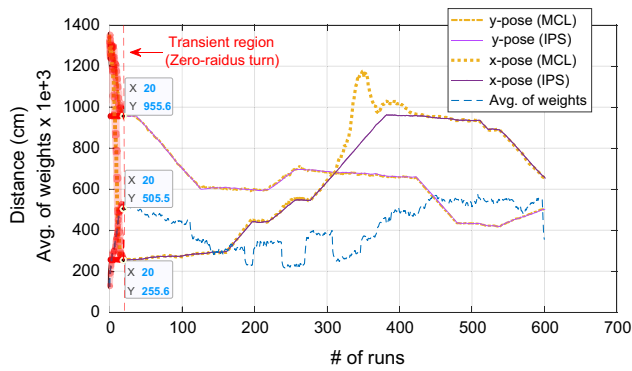


Fig. 25 Pose error analysis (Standard MCL)

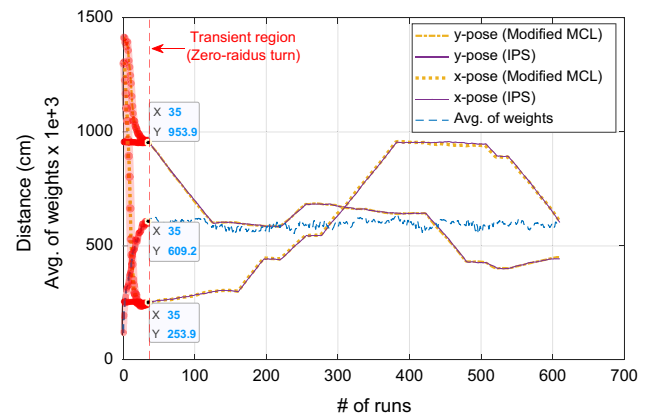


Fig. 27 Pose error analysis (Modified MCL)

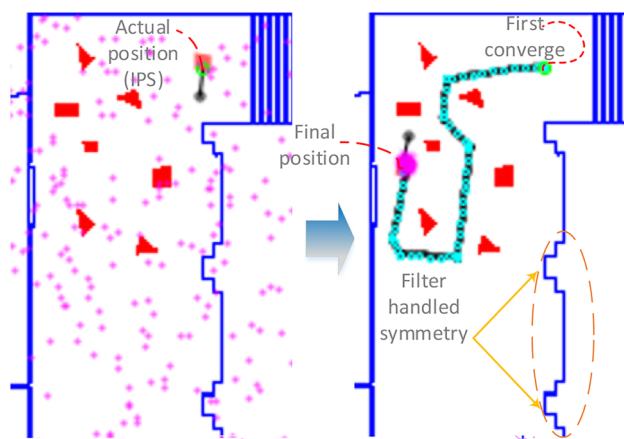


Fig. 26 Modified MCL results

its navigation. This phenomenon is depicted through the pose error, observed in both the x and y axes, as illustrated in Fig. 25. As can be seen on the chart, a substantial increase in pose error, particularly along the x -axis, is noted. This can be attributed to the heightened sensitivity of the filter to map symmetry when dynamic objects are introduced, resulting in alterations in filter convergence.

A new experiment employing the Modified Monte Carlo Localization (MCL) scheme was conducted while maintaining the same obstacle configuration. Due to the unpredictable effects resulting from physical conditions (e.g., wheel slippage, minor differences at initial convergence orientation), remote manipulation of the robot on the exact same trajectory could not be achieved. The trajectory for the Modified MCL scenario was obtained in the same form at the major level, with control inputs being sent to the robot in same order, as was done for the Standard MCL scenario. Figure 26 depicts the defined trajectory, received IPS measurements, and modified MCL position estimates as the Modified MCL scheme was applied.

Table 1 Experimental results for scenario with unmapped obstacle presence (750 particles)

	RMSE (x -axis)	RMSE (y -axis)	Weight (Mean of avg. of weights)
Standard MCL	66.65	4.17	0.4191
Modified MCL	4.47	2.61	0.5883

An increase in the reliability of position estimates has been noted when compared to the results obtained in the previous scenario. This enhancement can primarily be ascribed to the filter's more robust response to map symmetries, a result of the clamping strategy's implementation. Furthermore, the mitigation of excessive particle scattering in the presence of unmapped obstacles is effectively achieved through the rejection of laser beams that do not align with the static map and the utilization of ray casting data obtained from the most trusted particle. This approach is considered a novel and effective method within the field of resampling strategies.

The error graph, depicting the disparity between the robot's actual position and the position estimation provided by the proposed scheme, is presented in Fig. 27. It is evident from this graph that following the initial convergence period (after the transient region), a robust positioning estimation was achieved, aligning closely with the actual position along both the x and y axes.

Root mean squared error (RMSE) analysis of x -axis and y -axis of standard MCL and the proposed (modified) MCL is given in Table 1.

Table 2 presents the convergence levels achieved with varying particle counts (1500, 5000, and 10,000) for the same map. The results indicate an improvement in the position estimation performance of the filter as the number of particles increases. However, given the imperative of processing

Table 2 Convergence levels with varying particle counts

	1500 particles	5000 particles	10,000 particles
Standard MCL	0.4957	0.5221	0.6898
Modified MCL	0.6841	0.7833	0.8984

speed in real-time studies, it is more practical to operate with a manageable number of particles.

The experiments conducted in this study encompass an area of 15 m × 15 m, and satisfactory results were obtained using a minimum of 750 particles within this domain. To attain comparable performance, the standard MCL approach necessitates a substantially higher number of particles, leading to a significant increase in computational burden.

5 Conclusions

Implementing resampling strategies within the Monte Carlo framework introduces several potential limitations and challenges. One critical concern is the risk of sampling bias, where the chosen resampling method may inadvertently skew the representation of the underlying distribution, leading to inaccuracies in the simulation results. Additionally, the computational complexity associated with certain resampling strategies can pose challenges, particularly as the scale of the simulation increases. There's also a trade-off involving the loss of information, as the choice of resampling technique may result in the omission of valuable data, affecting the overall precision of the Monte Carlo simulation. Selecting the optimal resampling strategy becomes a critical decision, as an inappropriate choice may yield suboptimal results, emphasizing the need for careful consideration and evaluation in the application of these strategies to ensure the reliability and validity of simulation outcomes.

To overcome the inherent limitations of resampling strategies within the Monte Carlo framework, several alternative approaches can be implemented. First and foremost, bias correction techniques play a crucial role in mitigating the impact of sampling bias, ensuring a more accurate representation of the underlying distribution. To address the computational complexity associated with large-scale simulations, the adoption of efficient resampling algorithms, designed for parallel processing or optimized implementations, can significantly enhance computational efficiency. Furthermore, embracing adaptive resampling approaches that dynamically adjust to the evolving characteristics of the problem or system can improve adaptability. Hybrid resampling techniques, combining multiple strategies, or integrating other simulation

methodologies, offer a comprehensive and robust solution by leveraging the strengths of different approaches.

In this context, the resampling strategy proposed in this study is an approach to protect valuable data and increase the overall sensitivity of the Monte Carlo method and minimize information loss. As a future work, it is planned to make an improvement on the "kidnapped robot problem" for very highly dynamic environments.

Acknowledgements This study was conducted in the Machine Vision Laboratory of Kocaeli University Mechatronics Engineering department.

Author contributions S. Karakaya wrote the manuscript, prepared figures, and organize the experimental work.

Declarations

Conflict of interest The author has no relevant financial or non-financial interests to disclose. The author has no conflicts of interest to declare that are relevant to the content of this article. The author certifies that he has no affiliation with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The author has no financial or proprietary interests in any material discussed in this article.

References

- Dulimart HS, Jain AK (1997) Mobile robot localization in indoor environment. *Pattern Recogn* 30(1):99–111
- Gao X, Li J, Fan L, Zhou Q, Yin K, Jianxu W, Song C, Huang L, Wang Z (2018) Review of wheeled mobile robots' navigation problems and application prospects in agriculture. *IEEE Access*. 49248–49268.
- Baltzakis H, Trahanias PA (2003) Hybrid framework for mobile robot localization: Formulation using switching state-space models. *Auton Robot* 15:169–191
- Huang GP, Trawny N, Mourikis AI, Roumeliotis SI (2011) Observability-based consistent EKF estimators for multi-robot cooperative localization. *Auton Robot* 30:99–122
- Boccardo M, Martinelli F, Pagnottelli S (2010) Constrained and quantized Kalman filtering for an RFID robot localization problem. *Auton Robot* 29:235–251
- Munaro M, Menegatti E (2014) Fast RGB-D people tracking for service robots. *Auton Robot* 37:227–242
- Akai N (2023) Reliable Monte Carlo localization for mobile robots. *Journal of Field Robotics* 40:595–613
- Buchanan R, Bednarek J, Camurri M (2021) Navigating by touch: haptic Monte Carlo localization via geometric sensing and terrain classification. *Auton Robot* 45:843–857
- He S, Song T, Wang P et al (2023) An enhanced adaptive Monte Carlo Localization for service robots in dynamic and featureless environments. *J Intell Robot Syst* 108:6
- Peavy M, Kim P, Oyediran H, Kim K (2023) Integration of real-time semantic building map updating with Adaptive Monte Carlo Localization (AMCL) for robust indoor Mobile Robot Localization. *Appl Sci* 13(2):909
- Jaenal A, Moreno F-A, Gonzalez-Jimenez J (2023) *Int J Robot Res*. *Int J Robot Res* 42(13):1117–1132

12. Chen C, Ma Y, Lv J, Zhao X, Li L, Liu Y, Gao W (2023) OL-SLAM: a robust and versatile system of object localization and SLAM. *Sensors* 23(2):801
13. Facerias M, Puig V, Alcalá E (2022) Zonotopic linear parameter varying SLAM applied to autonomous vehicles. *Sensors* 22(10):3672
14. Perea D, Hernández-Aceituno J, Morell A, Toledo J, Hamilton A, Acosta L (2013) MCL with sensor fusion based on a weighting mechanism versus a particle generation approach. In: 16th international IEEE conference on intelligent transportation systems, Netherlands.
15. Patten T, Martens W, Fitch R (2018) Monte Carlo planning for active object classification. *Auton Robot* 42:391–421
16. Ge G, Zhang Y, Wang W, Jiang Q, Hu L, Wang Y (2011) Text-MCL: autonomous mobile robot localization in similar environment using text-level semantic information. *Machines* 10(3):1–21
17. Zhang L, Zapata R, Lépinay P (2009) Self-adaptive Monte Carlo localization for cooperative multi-robot systems. In: 10th Towards autonomous robotic systems (TAROS 2009), pp 259–266.
18. Nagi I, Adiprawita W, Mutijarsa K (2014) Vision-based Monte Carlo localization for RoboCup Humanoid Kid-Size League. In: 13th International conference on control automation robotics & vision (ICARCV), Singapore.
19. Karakaya S, Ocak H (2020) Low cost easy-to-install indoor positioning system. *J Intell Robot Syst* 100:131–144

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.