Transactions on Computational and Scientific Methods | Vo. 4, No. 8, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Enhanced Gmapping Algorithm Utilizing Improved Particle Swarm Optimization for Efficient SLAM Performance

Dominic Thorne
Department of Computer Science, University of Nebraska-Lincoln, USA
dominic.thorne72@unl.edu

Abstract: In recent decades, Simultaneous Localization and Mapping (SLAM) has emerged as a critical area of research within autonomous robotic systems, playing a fundamental role in navigation and environmental mapping. Traditional Rao-Blackwellized Particle Filter (RBPF)-based SLAM algorithms face challenges related to particle degradation and high computational cost, particularly as the number of particles increases. This paper addresses these issues by proposing an enhanced Gmapping algorithm based on improved particle swarm optimization (PSO). The proposed method introduces normal distribution and compression factors to optimize PSO convergence, improving localization accuracy while reducing the number of sampled particles. Experimental results demonstrate that the improved algorithm reduces localization error and enhances mapping accuracy, all while lowering hardware requirements. These findings highlight the potential for deploying this algorithm on cost-effective robotic platforms in future applications.

Keywords: Gmapping; Rate of Convergence; Simultaneous Localization and Mapping; PSO.

1. Introduction

All In the last decades, Simultaneous Localization and Mapping (SLAM) has been one of the most actively researched problems in autonomous robotic systems [1].SLAM algorithms refer to the motion of a robot in a completely unfamiliar situation, through the sensors it carries, to estimate its own position and to construct a map of the environment through which it passes.SLAM is a prerequisite for trajectory planning, trajectory tracking and automatic navigation of robots, thus making it an important component of the navigation system of mobile robots.SLAM has achieved good results in the last decades of technological development.

The common algorithms used for LiDAR SLAM map creation are mainly filter-based (EKF-SLAM, RBPF-SLAM and Fast-SLAM) algorithms and graph optimization-based (PTAM-SLAM, SD-SLAM, ORB-SLAM, SVOSLAM, RGBD-SLAM, etc.) algorithms [2]. The traditional RBPF algorithms often use particle filtering algorithms to evaluate the state of the robot.To estimate the robot's pose, the algorithm needs a lot of particles, and the more particles, the more accurate the calculation. However, as the number of particles rises, so does the computational difficulty [3-6]. In order to change the assessment of the robot state, the particle filtering algorithm gradually modifies the particle weights based on the measurement findings. Particle deterioration is unavoidable as the algorithm's iterations are multiplied, though, as there will be a greater proportion of heavier particles and a steady decline or even disappearance of the smaller ones.In order to change the assessment of the robot state, the particle filtering algorithm gradually modifies the particle weights based on the measurement findings.The classic RBPF-SLAM algorithm's core has been significantly enhanced in the literature [7], in other words, during the particle sampling stage, the bit-pose evaluation and verification are carried out utilizing line segment features to minimize the sample area. The approach is then updated in the weight update stage, which lowers the computational cost by using weights based on similarity ratio selection. However, the current algorithm's map generation process suffers from intermittent particle deterioration and somewhat frequent resampling time. In the literature [8], a Rao-Blackwellized particle filter-based SLAM solution was proposed to improve the proposal distribution by exploiting the independence between particles, extracting samples from the motion and optimizing them by scan matching. Adaptive resampling with smooth likelihood is used to avoid particle exhaustion in environments with large nested loops. However in the absence of data compression, it inevitably increases memory consumption and it increases more significantly as the number of particles increases.

Based on Gmapping and improved particle swarm optimization algorithms, this paper proposes a Gmapping based on improved particle swarm convergence, which can not only effectively reduce the number of particles sampled, but also optimize and improve the resampling strategy to increase the proportion of high-weight particles without affecting the diversity of retained particles, thus reducing the impact caused by particle degradation, which can generate more reliable robot poses and achieve the ultimate goal of improving the accuracy of localization and environmental mapping effects.

2. Gmapping Algorithm

2.1. The Traditional RBPF-SLAM Algorithm

The full SLAM problem defines SLAM as the a posteriori estimation of the entire path and map.

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

The equation where x denotes the robot state, m denotes the environment map, z denotes the sensor observations, u denotes the input control, and the subscript denotes the moment, is the potential trajectory of the robot over time [1, t] given the measurements and control data (usually the odometer). the Rao-Blackwellized particle filter is an effective method for solving the SLAM problem, where the posterior trajectory and mapping is defined by Equation :

$$p(x_{1:t}, m \mid z_{1:t}, m) \qquad p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) p(m \mid z_{1:t}, u_{1:t})$$

The result is obtained by computing the product of the two posterior probabilities. Here, the mapping posterior $p(m \mid z_{1:t}, u_{1:t})$ can be efficiently computed by treating it as a mapping problem with known poses. The positional posterior points $p(x_{1:t} \mid z_{1:t}, u_{1:t})$ can be estimated by means of a particle filter where each particle has its own map [9].

2.2. Improvements to the Proposed Distribution

This motion model has the benefit of being simple to calculate, but the proposed distribution comes in second place only to the ideal solution, which is most likely to happen if the robot is fitted with a laser rangefinder, especially since the sensor data is much more accurate than estimates of robot motion based on odometers [10]. The accuracy of the control volume estimated by the odometer is much less than that of the observation model built by the laser sensor, and to overcome this problem, the nearest sensor observation can be considered when generating the next generation of samples, and by integrating into the proposal, the sampling can be focused on meaningful regions of observation probability [11,12]. Therefore Grisetti et al. proposed equation to calculate a better proposal distribution.

$$p(x_t \mid m_{t\ \ 1}^{(i)}, x_{t\ \ 1}^{(i)}, z_t, u_t) \qquad \frac{p(z_t \mid m_{t\ 1}^{(i)}, x_t^{(i)}) p(x_t \mid x_{t\ 1}^{(i)}, u_t)}{p(z_t \mid m_{t\ 1}^{(i)}, x') p(x' \mid x_{t\ 1}^{(i)}, u_t) dx'}$$

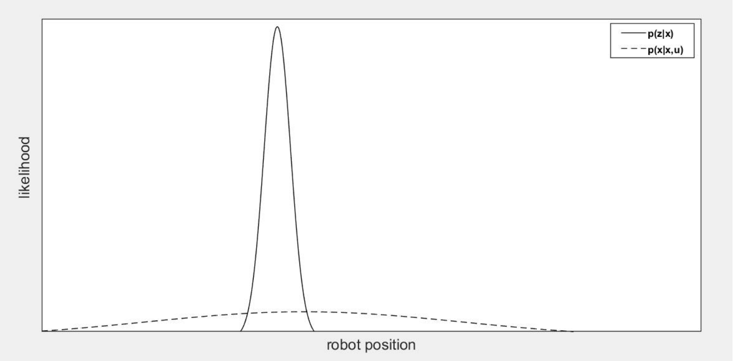


Figure 1. Distribution of Lidar observation model and odometer motion model

The dashed line is the probability distribution, which is a Gaussian distribution consistent with odometry sampling, and the solid line is the probability distribution, which is a Gaussian distribution of states obtained after observation using a laser. If the proposal distribution is represented by laser matching, the sampling interval can be restricted to a relatively small range, so that a smaller number of particles can be used to cover the probability distribution of the robot's poses, and for each particle a random scan can be matched to find the spatial location of the maximum in that region.

3. Experimental Results and Analysis

3.1. Improved PSO Algorithm Simulation

In order to evaluate the effectiveness of the improved PSO algorithm, simulations were carried out on MATLAB. The number of iterations in the simulation was set to 150, = 0.724 , c_1 = 1.49445 , c_2 = 1.49445, the population size was 50 and the expected value of the normal distribution was set to d = 0.5. The simulation results are shown in Figure 3. It can be seen that the improved PSO algorithm has fewer iterations compared to the traditional PSO, which proves to be faster in convergence.

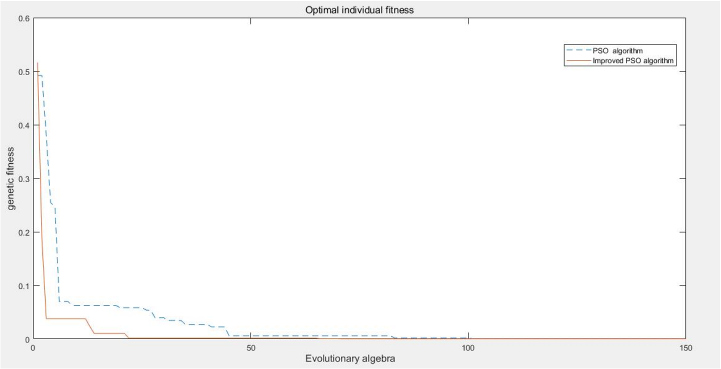


Figure 2. Comparison of improved PSO and traditional PSO iterative optimization

3.2. Positioning and Map Building

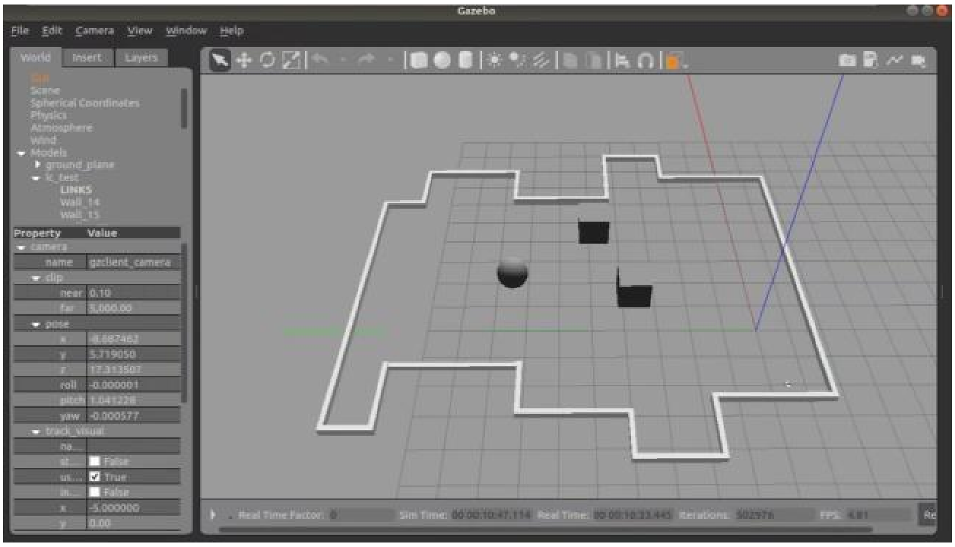
Table 1: Data comparison of the improved Gmapping

Algorithm	Datasets	Number of particles	Resampling times	Root Mean Square Error
Karto	FHW	14	112	0.204
	Freiburg Campus	36	131	0.294
Gmapping	FHW	14	128	0.189
	Freiburg Campus	36	157	0.256
Improved Gmapping	FHW	14	135	0.132
	Freiburg Campus	36	167	0.228

For the Gmapping algorithm incorporating the improved PSO, this paper first validates the simulation on two publicly available datasets, FHW and Freiburg Campus, on an Ubuntu 20.04 + ROS melodic environment. The simulation results and comparisons are shown in Table 1, where the mean square error RMSE is calculated from Equation, where X and X' are the actual and algorithm-located positions. From the data in Table 1, it can be seen that the number of resampling is higher compared to the Karto algorithm when the number of particles is maintained at the same level, but the RMSE after building the map results is significantly lower, proving that its building effect is more in line with the real environment.

$$RMSE \quad \sqrt{\frac{1}{n}\sum_{1}^{n}\left(X_k - X_k^{'}\right)^2}$$

Firstly, this paper uses the physical simulation platform Gazebo in ROS to add real-world physical simulation to the experiment, constructing a relatively simple environment model with the outer borders of the house white and square with simulated furniture and obstacles in the room, the simulation environment is constructed as shown in Figure 3.



Property	Value
camera	
name	gzclient_camera
clip	
near	0.10
far	5,000.00
pose	
x	-8.687462
y	5.719050
z	17.313507
roll	-0.000001
pitch	1.041228
yaw	-0.000577
track_visual	
na...	
st...	□ False
us...	☑ True
in...	□ False
x	-5.000000
y	0.00

Real Time Factor: 0 Sim Time: 00:00:10:47.114 Real Time: 00:00:10:33.445 Iterations: 502976 FPS: 4.81

Figure 3. Indoor simulation environment

The Gmapping algorithm is primarily used to check for consistency in the experimental data. In addition, a handle was utilized to control the movement of the AGV model in the simulated environment, which was used to gather data sets and record laser, odometer, and IMU data.

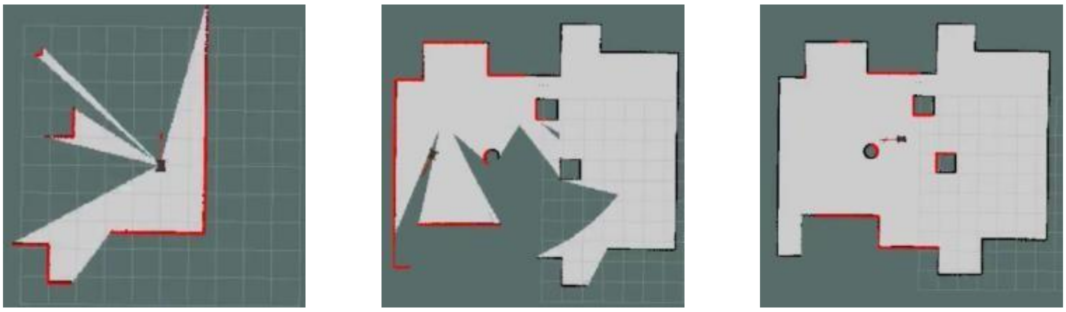


Figure 4. Built figure results of Gmapping

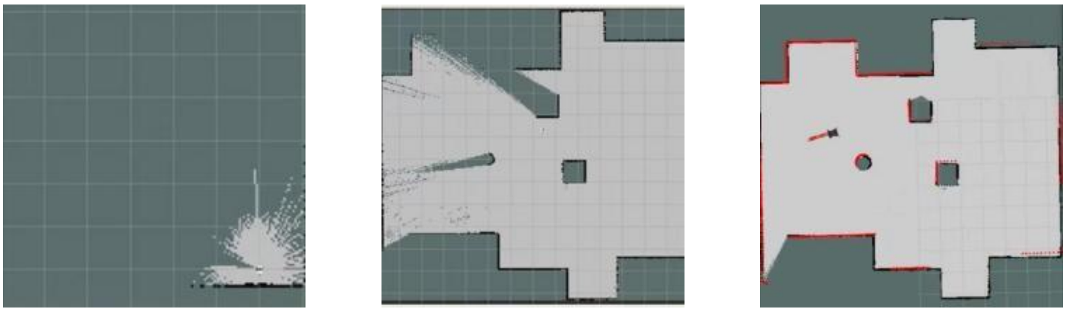


Figure 5. Built figure results of Karto

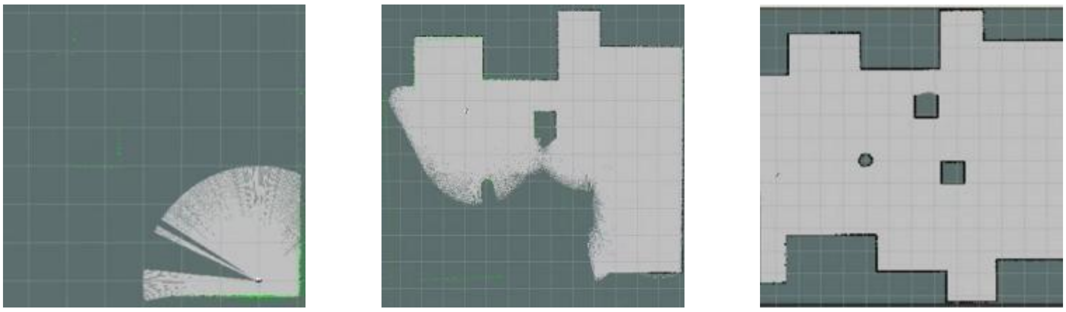


Figure 6. Built figure results of the improved

The results of the improved algorithm are compared to the current open source solution with Karto, which is recognised for its good results. The optimised algorithm achieves excellent build results with low sampled particles, and its overall convergence is much better and matches the actual map. Moreover, as only a low number of particles is required, the environmental configuration required to host the algorithm is reduced.

4. Conclusion

This paper introduces a normal distribution and a compression factor on the basis of the traditional PSO algorithm to improve the PSO convergence speed and the possible local optimum solution condition, based on which Gmapping based on the improved particle swarm convergence algorithm is proposed. Through experimental verification, the algorithm reduces the localization error and improves the algorithm's map building effect with a low number of sampled particles. At the same time low sampling particles require lower hardware configuration to run the algorithm, which is beneficial to invest in lower cost hardware in the future.

References

[1] Fernando Martin, Luis Moreno, Dolores Blanco. Kullback-Leibler divergence-based global localization for mobile robots[J]. Robotics & Autonomous Systems, 2014 (62): 120-130.

[2] Li, Ghaemidizaji M, Dadkhah C, Leung H. Efficient robot localization and SLAM algorithms using Opposition based High Dimensional optimization Algorithm[J]. Engineering Applications of Artificial Intelligence: The International Journal of Intelligent Real-Time Automation, 2021(104-):104. J. Liu, E.L. Chen and Z.T. He: Journal of Shi Jia Zhuang Railway Institute (Natural Science), Vol. 22 (2009) No. 4, p.40-42.

[3] Qu Liping, Wang Hongjian. An overview of robot SLAM problem[C]. IEEE International Conference on Consumer Electronics, 2011:1953-1956.

[4] Liu Juncheng, Li Di, Weng Xiaowen. Research on Mobile Robot SLAM Based on Laser Information[J]. Automation and Instrumentation, 2018, 33(6): 43-47.

[5] Luo Yuan, Yu Jiahang, Wang Longfeng, Wang Yunkai. Simultaneous localization and mapping of an improved RBPF based mobile robot[J]. CAAI Transactions on Intelligent Systems, 2015, 10(3):460-464.

[6] Zheng Bin, Chen Shili, Liu Rong. Research on gmapping based on firefly algorithm optimization[J]. Computer Engineering, 2018, 44(9):22-27.

[7] Shen Y, Jiao Z. A Novel Self-Positioning Based on Feature Map Creation and Laser Location Method for RBPF-SLAM[J]. Journal of Robotics, 2021, 2021.

[8] CAO Pedrosa E, Pereira A, Lau N. Fast grid slam based on particle filter with scan matching and multithreading [C]//2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, 2020: 194-199.

[9] GRISETTI G., STACHNISS C. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters [J]. IEEE Transactions on Robotics: A publication of the IEEE Robotics and Automation Society, 2007, 23(0):34-46.

[10] Zhou L, Koppel D, Kaess M. LiDAR SLAM with plane adjustment for indoor environment[J]. IEEE Robotics and Automation Letters, 2021, 6(4): 7073-7080.

[11] Pedrosa E. Fast grid slam based on particle filter with scan matching and multithreading[C]//2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, 2020: 194-199.

[12] Chen S, Zhou B, Jiang C, et al. A LiDAR/Visual SLAM Backend with Loop Closure Detection and Graph Optimization[J]. Remote Sensing, 2021, 13(14): 2720.

[13] Rossides G, Hunter A, Metcalfe B. Source Localisation Using Wavefield Correlation-Enhanced Particle Swarm Optimisation[J]. Robotics, 2022, 11(2): 52.

[14] Song H, Zhao Z, et al. Research on SLAM Algorithm of Mobile Robot Vision Based on Deep Learning[C]//2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing). IEEE, 2021: 1-7.