https://doi.org/10.1007/s12065-020-00545-y

RESEARCH PAPER

Particle swarm optimization for solving a scan-matching problem based on the normal distributions transform

Sara Bouraine[1] · Abdelhak Bougouffa[1] · Ouahiba Azouaoui[1]

Abstract
In this paper, an evolutionary scan-matching approach is proposed to solve an optimization issue in simultaneous localization and mapping (SLAM). A rich literature has been invested in this direction, however, most of the proposed approaches lack fast convergence and simplicity regarding the optimization process, which should directly affect the accuracy of the environment's map and the estimated pose. It is a line of research that is always active, offering various solutions to this issue. Among many SLAM methods, the normal distributions transform approach (NDT) has shown high performances, where numerous works have been published up to date and many studies demonstrate its efficiency *wrt* other methods. Nevertheless, few works have been interested to solve the optimization issue. The proposed solution is based on NDT scan-matching using particle swarm optimization (PSO) and it is dubbed NDT-PSO. The main contribution is to solve the pose estimation problem based on PSO and iterative NDT maps. The performances of the NDT-PSO approach have been proven in real experiments performed on a car-like mobile robot in both static and dynamic environments. NDT-PSO is tested for different swarm sizes, and the results show that 70 particles are more than enough to find the best particle while avoiding local minima even in loop closing. The algorithm is also suitable for real time applications, with an average runnnig time of 145ms for 70 particles and 70 iterations of the optimization process. This value can be further reduced using fewer particles and iterations. The accuracy of the proposed approach is also evaluated *wrt* other SLAM methods widely used among the robot operating system community and it has been shown that NDT-PSO outperforms these algorithms.

Keywords Simultaneous localization and mapping · Scan matching · Particle swarm optimization · Normal distributions transform · Mobile robots · Bio-inspired behaviour · Pose estimation · Robot operating system

1 Introduction

Today, mobile robots are present in our daily life and they become increasingly autonomous. Indeed, thanks to the ability to locate and map its environment, the robot can plan trajectories and navigate in the real world in order to perform various tasks without human intervention. This ability is known as the simultaneous localization and mapping (SLAM) problem [1–6]. It consists of estimating the robot position while building a map of the environment. Most developed SLAM approaches are under the Filtering [7–10] or scan-matching [11–14] paradigms but share the same main steps, namely; mapping and pose estimation. The improvement of the earliest approaches in the two categories is still ongoing despite their existence for more than two decades. In this paper, the optimization problem is addressed in particular to solve a SLAM problem. There is a rich literature on optimization algorithms [15–17], however, evolutionary algorithms [18] present better performances to solve this issue, such as artificial bee colony (ABC) algorithm [19], sparrow search algorithm (SSA) [20], firefly algorithm (FA) [21], invasive weed optimization (IWO) [22], world cup optimization (WCO) algorithm [23], and particle swarm optimization (PSO) [24]. Among these methods, PSO is one of the most effective algorithms, due to its fast convergence and its ability to find a global optimum [25].

✉ Sara Bouraine
s_bouraine@yahoo.fr

Abdelhak Bougouffa
abdelhak.bougouffa@universite-paris-saclay.fr

Ouahiba Azouaoui
oazouaoui@cdta.dz

[1] Center for development of advanced technologies (CDTA), Cite 20 Aout 1956, Baba Hassen, Algiers, Algeria

Indeed, in recent years, particle swarm optimization is increasingly exploited in SLAM-based methods to eliminate the problems of insufficiency and inaccuracy of prior information for Kalman filter methods [26] and impoverishment of particles for particle filter methods [27–29]. In [26], the fuzzy adaptive extended Kalman filtering method has been improved by introducing the fractional-order Darwinian particle swarm optimization (PSO) to compute an accurate prior noise model. Lee et al. [27] proposed a FastSLAM framework where the robot position is estimated using a Rao-Blackwellized particle filter. The accuracy of the Fast-SLAM deteriorates over time due to particles depletion in the re-sampling phase. To prevent degeneration, a particle swarm optimization is employed to solve the problem by means of particle cooperation. Another improvement of the Fast-SLAM is the work of Liu et al. [28] where PSO and unscented particle filters are introduced to reduce drastically the number of particles thanks to PSO for pose estimation and to improve map estimation accuracy thanks to the unscented particle filter. In Wu et al. [29], to solve the degeneration of particles and the positioning inaccuracy (due to the need for a large number of particles) problems, the Gaussian particle swarm optimization algorithm is used in the particle filter process.

In the second category, namely scan-matching methods, most of the methods use a nonlinear least-square optimization to determine the robot pose, but it is subject to the local minima problem which leads to not guaranteeing the algorithm convergence particularly in the presence of dynamic objects or fast movement of the robot. These few last years, some works have been conducted to prevent and improve the scan-matching-based methods. In [14], the ICP and Hector SLAM algorithms are improved using the system model as an initialization step followed by the ICP or Hector as an alignment step. The work in [11] introduces in the ICP algorithm both a 2D laser scan-matching method using point and line features as an initialization phase and the lq-norm ($0 < q < 1$) metric as a pose estimation to filter the outliers. In Wang et al. [12], a mixture of exponential power (MoEP) distributions is used for approximating the residual error distribution. The optimization of the scan-matching method is iteratively achieved via two phases: an on-line parameter learning (OPL) phase to learn residual error distribution for a better representation according to the likelihood field model (LFM), and an iteratively re-weighted least squares (IRLS) phase to attain transformation for accuracy and efficiency. Nevertheless, among many SLAM methods, NDT has shown high performances, where numerous works have been published up to date [30–36, 38] also many studies demonstrate its efficiency *wrt* other methods [39–42]. However, to solve the optimization problem, most NDT-based methods use Newton algorithm (like [43, 44]). Few works have been interested in the problem of optimization; in [38, 41], the pose estimation is performed using Monte Carlo Localization (MCL). In [34, 45], the best fitting alignment between two point samples sets is found through the minimization of the L2 distance between NDT models. Moreover, in other works, the optimal transformation to solve the scan-matching problem is directly formulated as a maximum likelihood estimate of Gaussian mixture maps [36].

The main contribution of this paper is to propose a new NDT-based scan-matching method dubbed NDT-PSO. Among SLAM methods and NDT-based methods in particular, few works have been interested in the problem of optimization in pose estimation. Most approaches use gradient-based approaches such as the least squares or Newton methods, known to be be computationally expensive and sensitive to the choice of the departure position affecting the convergence of the algorithm. However, the proposed approach is based on PSO, where to the authors' knowledge this is the first time that it is used to solve the problem of pose estimation in the SLAM issue. Furthermore, this paper adds a new approach to the NDT's variant family, which overcomes the cited problems.

The principle advantages of NDT-PSO are its efficiency, its fast convergence, and its ability to find a global optimum. Moreover, the problems of insufficiency and inaccuracy of prior information do not arise, as is the case for the filtering methods. Another significant advantage of NDT-PSO is that dynamic objects do not affect the accuracy of the results without the need for additional algorithms. NDT-PSO is also very suitable for real time applications, which is a worry in most evolutionary optimization techniques.

The highlights of the research can be expressed as follows: (1) NDT-PSO is based on a normal distributions transform (NDT) for the environment mapping. This approach overcomes uncertainties and environment constraints. (2) To determine the estimated position of the robot, the optimization problem in the scan-matching process is solved using a particle swarm optimization approach. It is a key issue in scan-matching-based SLAM methods. The solution is encoded as the best particle in the swarm, representing the best transformation between two successive NDT maps. (3) NDT-PSO is based on a modified version of PSO that incorporates an inertia weight [37]. (4) It has been tested in real experiment to validate the algorithm in real world conditions on a car-like mobile robot. Experiments have been carried out in both static and dynamic environments. (5) NDT-PSO has been also evaluated *wrt* other SLAM methods widely used among the ROS community, namely gmapping [46] and Hector-SLAM [47] methods. The obtained results show a high accuracy of NDT-PSO compared to the two other methods.

This paper is organized as follows: Sect. 2 illustrates the proposed approach, where the scan-matching problem is formalized before presenting the NDT-PSO algorithm.

Validations in real experiments are given in Sect. 3. Finally, Sect. 4 provides a summary and a conclusion of the paper.

2 Proposed approach: NDT-PSO

To solve an optimization problem in simultaneous localization and mapping (SLAM), the proposed solution is based on a bio-inspired approach dubbed normal distributions transform particle swarm optimization (NDT-PSO). Like any SLAM problem, two key issues to answer are environment mapping and pose estimation. The following sections answer these two issues.

2.1 Environment representation

In this paper, a normal distributions transform (NDT) based representation is used to deal with uncertainties and environment constraints (presence of objects with arbitrary shapes). Proposed by Biber and Strasser [30], NDT is a refined version of the popular occupancy grid approach [48, 49]. As a common point, NDT is based upon a grid discretization of the space; the collected data for a given time step (a 2D laser scan in the case of this paper) are modeled as a set of cells with an appropriate resolution. However, by taking a closer look to a cell, instead of affecting a probability of occupancy to the whole cell, a probability of measurement is associated with each 2D scan point in this cell.

Let $M = \{q_j\}, j = 1, \ldots, N_M$ a points set corresponding to a given laser scan, with $q_j \in \mathbb{R}^2$. These raw data are transformed to NDT-based representation by the following steps:

- As the space is subdivided into a set of cells, each cell with ID "c" will be assigned a sub-set of points m_c containing N_{m_c} points, i.e. $m_c = \{q_i\}, i = 1, \ldots, N_{m_c}$, where $m_c \subset M$.
- For each grid cell (with ID c), the two parameters μ_c and Ω_c representing respectively the mean and the covariance should be computed to determine the corresponding normal distribution $N(\mu_c, \Omega_c)$. They have the form:

$$\mu_c = 1/N_{m_c} \sum_{i=1}^{N_{m_c}} q_i \tag{1}$$

$$\Omega_c = 1/N_{m_c} \sum_{i=1}^{N_{m_c}} (q_i - \mu_c)(q_i - \mu_c)^t \tag{2}$$

with $\mu_c \in \mathbb{R}^2$ and $\Omega_c \in \mathbb{R}^2 \times \mathbb{R}^2$

The NDT map is therefore represented by a set of local normal distributions.

2.2 Pose estimation

From a SLAM perspective, the position is computed based on a scan-matching approach with the basic principle of comparing two successive scans. By founding the transformation between the two scans, the position is estimated as long as this transformation represents the robot displacement. To this end, an optimization approach should be invested to solve the problem of pose estimation. Standard SLAM approaches, including NDT use gradient-based approaches such as Newton method [50] with major drawbacks to be computationally expensive and sensitive to the choice of the departure position, i.e. if this position is far from the real solution, the method can either diverge or converge to a local minimum. It is a frequent problem in scan-matching [51, 52]. In this paper, the proposed approach NDT-PSO is based on the particle swarm optimization approach (PSO), which is commonly used to solve several optimization problems due to its efficiency and simplicity. Furthermore, PSO is more likely to fall on the global minimum without position initialization [53, 54]. An important contribution in this work is the exploitation of particle swarm optimization for solving a SLAM problem.

PSO is a stochastic population-based approach developed by Kennedy (a social psychologist) and Eberhart (an electrical engineer) who combined their expertise to model and formalize an animal behaviour based on fish schooling and bird flocking [55]. The swarm movement is a very intelligent behaviour, which can be simply interpreted as follows:

During their motion, each individual of the population has to continuously update its position given its own experience over time and the experience of its neighbours, with the ultimate goal of reaching food.

This behaviour is similar to an optimization problem where each individual is a possible solution dubbed particle.

Let p a particle defined by its position $X \in \mathbb{R}^D$ and velocity $V \in \mathbb{R}^D$ with D the dimension of the search space. Based on a modified version of PSO proposed by Shi and Eberhart [37], the problem can be formalized thanks to the following equation:

$$V_{\tau+1}(p) = \sum_n F_n(p, w_n) \tag{3}$$

From Eq. 3, $V_{\tau+1}(p)$ is the velocity of p at the iteration $\tau + 1$ of the optimization process expressed according to the functions $F_n, n = 1, ..., 3$ representing different attractive forces affecting the motion of the particle and the parameters $w_n, n = 1, ..., 3$ depicting weighting factors balancing the importance of each force. The first function represents the momentum behaviour expressed by forces attracting the particle to keep its current motion and has the form:

$$F_m = w_m V_\tau(p) \tag{4}$$

with w_m the momentum weighting factor (also called the inertia weight) and V_τ particle's current velocity.

The second function F_c represents the cognitive behaviour depicting forces constraining the particle to consider its own experience by biasing its motion toward the personal best position denoted *Pbest*. It has the form:

$$F_c = w_c |rand_1| (Pbest_\tau(p) - X_\tau(p)) \tag{5}$$

where *rand*1 is a random variable uniformly distributed in the range [0, 1], w_c is the cognitive weighting factor and X_τ is the current position of the particle.

The third function F_s concerns the social behaviour where the particle considers the swarm's experience by adjusting its motion according to the global best position denoted *Gbest*. It represents the best position found so far in the swarm, such that:

$$F_s = w_s |rand_2| (Gbest_\tau - X_\tau(p)) \tag{6}$$

with *rand*2 is a random variable and w_s is the social weighting factor. w_c and w_s are also known as acceleration coefficients. The velocity of each particle in the swarm is updated thanks to equation 3, which in turn is used to update the particle's position according to the following equation:

$$X_{\tau+1}(p) = X_\tau(p) + V_{\tau+1}(p) \tag{7}$$

In NDT-PSO, pose estimation problem is solved by encoding the geometric transformation T (translation (T_x, T_y) and rotation θ) between two scans into a particle:

$$X(p) = T = (T_x, T_y, \theta) \tag{8}$$

Let M_{k-1} and M_k two successive scans at iterations $k-1$ and k. A 2D point $q_j \in M_k$ can be represented in the coordinate frame of the scan M_{k-1} thanks to:

$$q_j' = T(q_j, X) \tag{9}$$

where T depicts the geometric transformation between two robot Cartesian coordinate frames, such that:

$$\begin{pmatrix} q_{xj}' \\ q_{yj}' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} q_{xj} \\ q_{yj} \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} \tag{10}$$

The swarm particles ($p = 1, \ldots, N_p$) propose possible solutions to the scan-matching problem. In our case, the optimal solution should be selected with respect to the best matching between the two scans M_{k-1} and M_k. Therefore, it is evaluated by summing the normal distributions P of all points q_j' given the transformation expressed by the vector $X(p)$. Therefore, the particles are optimized by maximizing the following objective function:

$$f(p) = \sum_{j=1}^{N_M} P(q_j') \tag{11}$$

The normal distribution $P(q_j')$ of each mapped point q_j' is determined by a simple lookup in the built map MAP_{k-1}. After attributing each scan point q_j' to its corresponding cell c, the measurement probability P of q_j' is computed thanks to:

$$P(q_j') = \exp(-(q_j' - \mu_c)^t \Omega_c^{-1} (q_j' - \mu_c)/2) \tag{12}$$

2.3 NDT-PSO algorithm

From the NDT-PSO approach illustrated above, two key aspects can be highlighted: (1) the environment representation is based on the normal distributions transform approach and (2) the scan-matching is based on a bio-inspired stochastic approach. Algorithm 1 depicts the NDT-PSO process for a given iteration k. Based on two successive scans M_{k-1} and M_k (respectively at previous and current iteration) and given a particles swarm with N_p population size. NDT-PSO computes the geometric transformation Tran between M_{k-1} and M_k and accordingly updates environment's global map GMAP. These parameters are respectively inputs and outputs of the algorithm. In the first step of the algorithm, the scan M_{k-1} is mapped given Eqs. 1 and 2 as illustrated in Sect. 2.1 (thanks to the function BUILD_MAP). It results local normal distributions based map denoted MAP_{k-1} (line # 1, algorithm 1, which on this basis will compute the transformation Tran. The next algorithm step is the swarm initialization corresponding to the first optimization process iteration ($\tau = 0$). During this step, the whole particles ($p = 1, \ldots, N_p$)) are randomly initialized according to the function RANDOM_INITIALIZATION, where each particle is defined by its position vector X_τ and velocity vector V_τ. Based on $X_{\tau=0}$, the personnel best $Pbest_\tau$ is initialized and the set of scan points M_k is mapped according to the function MAPPING_SCAN. This function proceeds mainly in two steps; (1) Determine the transformation of the set of points $q_j \in M_k$ into the coordinate frame of scan M_{k-1} according to Eq. 10 (line # 37, algorithm 1. (2) Compute the normal distribution P of each mapped point q_j' according to Eq. 12 (line # 38, algorithm 1. Based on $P(q_j')$ of the whole scan, the objective function f_τ for each particle is evaluated and consequently the global best particle $Gbest_\tau$ is determined. Given $Pbest_\tau$ and $Gbest_\tau$, the velocity and position vectors ($X_{\tau+1}$ and $V_{\tau+1}$) are updated (lines # 13 and # 17, algorithm 1. Therefore, the scan points $q_j \in M_k$ are re-mapped according to the updated vector $X_{\tau+1}$ (line # 18, algorithm 1. Similarly to the first step of the optimization process, the objective function $f_{\tau+1}$ is computed based on updated parameters. Given the maximization criterion, the

personnel best $Pbest_{\tau+1}$ and the global best $Gbest_{\tau+1}$ are computed. This process is repeated for a maximum number of iterations $iteration_{max}$. At the end of the process, the best solution corresponding to the global best (the best found so far) is assigned to the transformation *Tran*, and correspondingly, the global map GMAP is updated with M'_k, such that $M'_k = \{q_j' \in M'_k / q_j' = T(q_j, Tran) \wedge Tran = Gbest\}$.

Algorithm 1 NDT-PSO process.

Input: M_{k-1} (laser scan corresponding to previous iteration), M_k (laser scan corresponding to current iteration), number of particles N_p.
Output: $Tran$ geometric transformation between two scans (at k and $k+1$), GMAP.

1: $\text{MAP}_{k-1} \leftarrow$ BUILD_MAP(M_{k-1}); //Map construction given equations 1 and 2
2: //Particles initialization (optimization iteration $\tau = 0$)
3: **for** each particle p=1 to N_p **do**
4: $(X_\tau(p), V_\tau(p)) \leftarrow$ RANDOM_INITIALIZATION();
5: $Pbest_\tau(p) = X_\tau(p)$;
6: MAPPING_SCAN(M_k,$X_\tau(p)$);
7: compute $f_\tau(p)$; // the objective function according to equation 11
8: **end for**
9: $Gbest_{index} = \underset{p=1,\ldots,N_p}{\text{argmax}}\ f_\tau(p)$; //index of the particle corresponding to the global best
10: $Gbest_\tau = X_\tau(Gbest_{index})$;
11: **while** $\tau < iteration_{max}$ **do**
12: **for** each particle p=1 to N_p **do**
13: $V_{\tau+1}(p) = V_\tau(p) + w_c|rand_1|(Pbest_\tau(p) - X_\tau(p)) + w_s|rand_2|(Gbest_\tau - X_\tau(p))$;
14: **if** $|V_{\tau+1}(p)| > V_{pmax}$ **then**
15: $V_{\tau+1}(p) = V_{pmax}$;
16: **end if**
17: $X_{\tau+1}(p) = X_\tau(p) + V_{\tau+1}(p)$;
18: MAPPING_SCAN(M_k,$X_{\tau+1}(p)$);
19: compute $f_{\tau+1}(p)$; // the objective function according to equation 11
20: **if** $f_{\tau+1}(p) > f_\tau(p)$ **then**
21: $Pbest_{\tau+1}(p) = X_{\tau+1}(p)$; //update personal best
22: **end if**
23: **end for**
24: //update global best
25: **if** $\max_{p=1,\ldots,N_p} f_{\tau+1}(p) > \max_{p=1,\ldots,N_p} f_\tau(p)$ **then**
26: $Gbest_{index} = \underset{p=1,\ldots,N_p}{\text{argmax}}\ f_{\tau+1}(p)$;
27: $Gbest_{\tau+1} = X_{\tau+1}(Gbest_{index})$;
28: **end if**
29: **end while**
30: $Tran = Gbest_{\tau+1}$;
31: $\text{GMAP} \leftarrow$ UPDATE_GLOBAL_MAP($Tran$,M_k);
32: **return** $Tran$,GMAP
33:
34: **Procedure** MAPPING_SCAN(M_k,X)
35: **for** each 2D point $q_j \in M_k$ **do**
36: //Transform q_j into coordinate frame of the scan M_{k-1} using equation 10
37: $q_j' = T(q_j, X)$;
38: Compute $P(q_j')$; //The normal distribution of q_j' according to equation 12
39: **end for**
40: **EndProcedure**

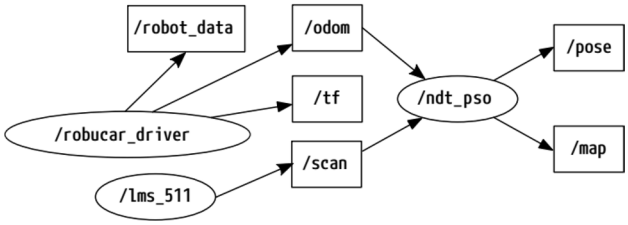

Fig. 1 ROS Computation graph generated by rqt_graph. It highlights ROS nodes and topics. The ndt_pso node communicates with the topics/laser and/odom (to respectively get the laser data and initialize the robot's position). This node is publishing the robot's position (via/pose topic) and the built map (via/map topic)

Fig. 2 Experimental platform Robucar

3 Results

To demonstrate and validate the performances of NDT-PSO algorithm, it has been implemented and tested on an experimental platform in static and dynamic, indoor and outdoor environments. The implementation has been done under the operating system ROS[1] (Robot Operating System) using C++ language. the ROS Computation graph of the NDT-PSO algorithm is depicted in figure 1.

3.1 Robot model

The experimental platform is a standard car-like vehicle called *Robucar*, with two fixed rear wheels and two orientable front wheels (see Fig. 2), governed by the following kinematic model:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = v \begin{bmatrix} \cos\phi \\ \sin\phi \\ \tan\xi/L \end{bmatrix} \tag{13}$$

with (x, y) the coordinates of the rear axle midpoint, ϕ the orientation of the robot, ξ the steering angle (orientation of the front wheels), v the linear velocity and L denotes the robot's wheelbase.

The *Robucar* is equipped with a laser range finder LMS511 placed in front of the robot, with an 80M maximum range and 190° field of view.

3.2 NDT-PSO at work

To illustrate how NDT-PSO works, it has been tested in different scenarios; the first one is called *CDTA-hall-scenario*, it shows the performances of NDT-PSO in indoor structured environment. The second scenario is called *CDTA-urban-scenario* which is more challenging given its unstructured nature. NDT-PSO has been also evaluated with respect to the presence of moving objects. The swarm size is set to 70 particles, the maximum number of optimization process iterations is 70. The acceleration coefficients in equation 5 and 6 are defined by $w_c = w_s = 2$, given [37, 55]. The particles are randomly initialized, however, for more accuracy, they are generated in a limited area around the previously estimated pose such that :

$X(p) \in [-\Delta x/2, \Delta x/2] \times [-\Delta y/2, \Delta y/2] \times [-\Delta\theta/2, \Delta\theta/2]$ $= [-1m, 1m] \times [-1m, 1m] \times [-\pi/8, \pi/8]$. With $\Delta x, \Delta y, \Delta\theta$; the dimensions of the initialization area given T_x, T_y translation and θ rotation.

NDT-PSO is based on a modified version of PSO that incorporates an inertia weight [37], which has been chosen after implementating and testing several PSO variants. The variants involved are : (1) A modified version of PSO that incorporate a constriction coefficient [56, 57]. (2) The global-local best PSO (GLBest-PSO), an improved version of PSO algorithm incorporating global-local best inertia weight (GLBest IW) with global-local best acceleration coefficient (GLBest Ac) [58]. (3) Rotation PSO (RPSO), in which the velocity vector of each particle is multiplied by a random rotation matrix at each iteration [59]. (4) The global-local optimal information ratio PSO (GLIR-PSO), which uses the global and local optimal information ratio to enhance appropriately the ability of global search [60]. (5) A modified PSO algorithm with a dynamic population size. It adjusts the size of the population for each iteration [61]. However, in our case, all these variants made no contribution *wrt* the modified PSO version used in this paper.

[1] www.ros.org

Fig. 3 NDT-PSO at work in *the CDTA hall scenario*; **a** CDTA hall from different view angles. **b** CDTA hall plan (the red trajectory is generated by NDT-PSO). **c** NDT-PSO generated map and trajectory for 70 particles and 1m map resolution

Fig. 4 CDTA urban scenario

3.2.1 CDTA hall scenario

This dataset has been recorded in *the CDTA hall scenario*, an indoor environment with static objects of different natures (see Fig. 3a and Fig. 3b). The resulting map and the robot's trajectory are depicted in Fig. 3c, where the quality of the built map and the accuracy of the estimated positions are high even in loop closing. The map has been faithfully rebuilt and the robot's trajectory correctly estimated.

3.2.2 CDTA urban scenario

The second scenario has been carried out in an urban environment at CDTA (represented in Fig. 4). It is an unstructured outdoor environment with the presence of arbitrary

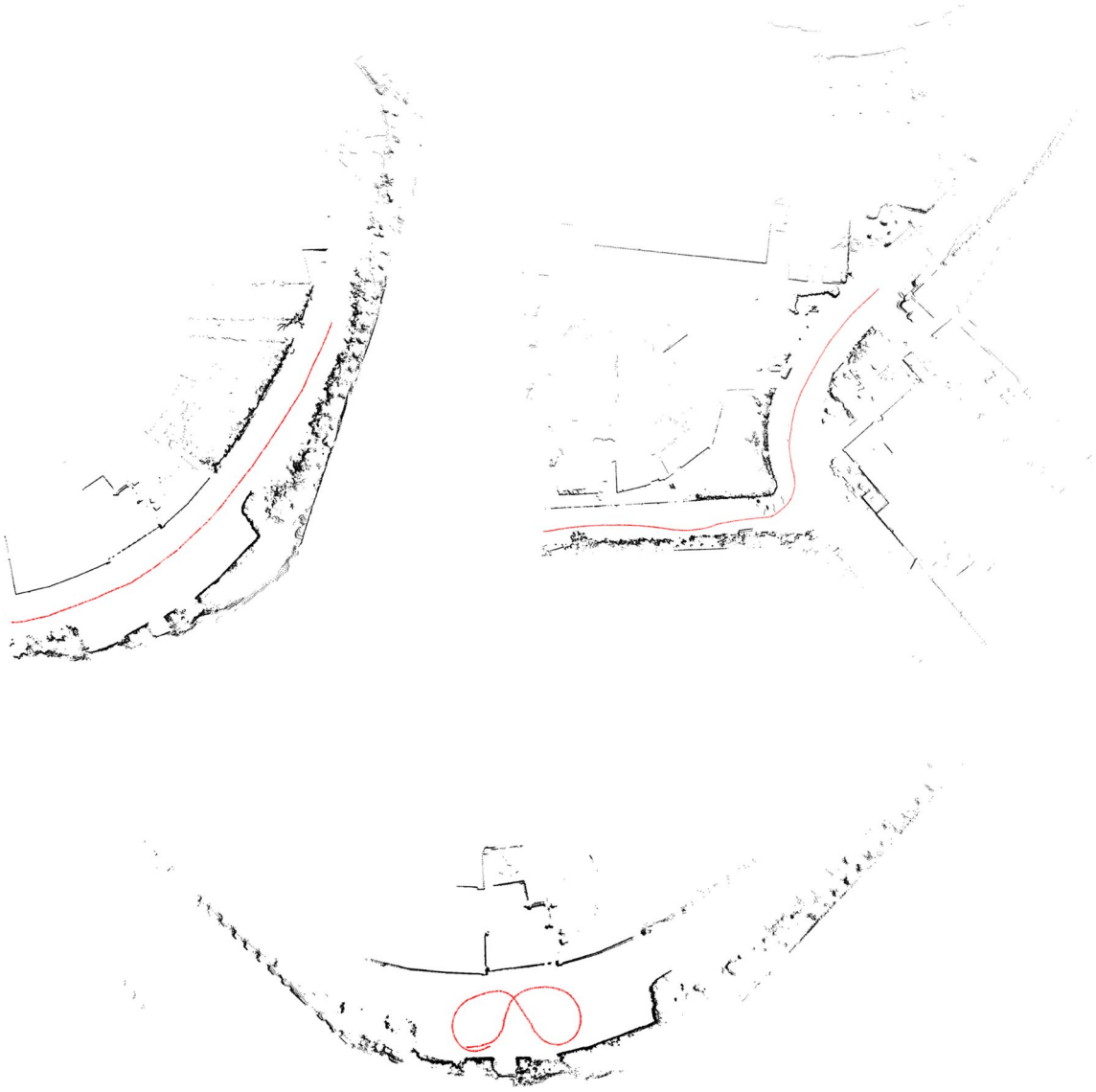

Fig. 5 NDT-PSO at work in *the CDTA urban scenario* for different situations (for 70 particles and 1M map resolution)

shaped objects (ex. trees, shrubs, buildings, persons, etc.). Figure 5 illustrates the resulting map and estimated trajectory for different robot paths. These experiments show the performance of NDT-PSO in an outdoor environment.

From an optimization perspective, NDT-based representation is known for its high capacities in such conditions as it deals with data uncertainty due to its probabilistic nature. Furthermore, with only 70 particles (where many tests were successful with less particles), NDT-PSO can find the best particle while avoiding local minima even in a loop closing situation (see Fig. 5 at the bottom).

To more illustrate NDT-PSO performances when closing loops, in the case of Fig. 6, the robot is performing three loops while maintaining a constant steering angle (for more clarity). The resulting robot trajectories seem perfectly superposed.

In figure 7, NDT-PSO has been tested in a more challenging environment conditions, so as to evaluate how it tackles moving objects with arbitrary trajectories. The experiments carry out in *CDTA urban scenario* with the presence of four pedestrians moving arbitrary in different directions. The environment and the robot's displacement are respectively correctly mapped and estimated. Objects movement appears clearly in the figure by the blue traces. It can be concluded that, the accuracy of the results is not affected by the presence of moving objects.

3.3 Algorithm performances

NDT-PSO algorithm is a function of some parameters; mainly the swarm size N_p and the number of optimization

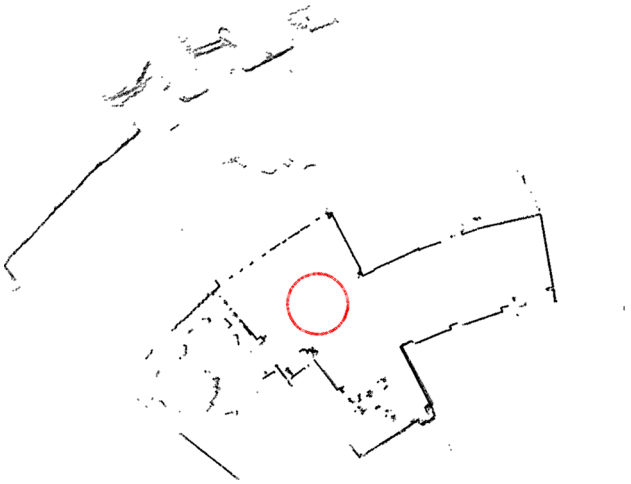

Fig. 6 NDT-PSO at work in a loop closing situation. The robot is performing three superposed circular trajectories (trajectories in red)

Fig. 7 NDT-PSO at work in the presence of moving objects in *the CDTA urban scenario* (see text)

Table 1 Average running time of NDT-PSO *wrt* the number of particles in the swarm (N_p), for $iteration_{max} = 70$ (tests carried out in figure 7)

Population size (N_p)	Running time (s)
10	0.0303893232
20	0.0487240912
30	0.0690673167
40	0.0864106667
50	0.1100520787
60	0.1250710661
70	0.1440125
80	0.1619612011
90	0.1784725106
100	0.1990441554

process iterations $iteration_{max}$. In the above experiments, these parameters have been set to fixed values, however, to more understand their effects on NDT-PSO, the algorithm has been assessed accordingly. The Tables 1 and 2 give respectively running times of NDT-PSO wrt N_p and $iteration_{max}$. From the obtained results, it is clear that the computation time complexity grows linearly with N_p and $iteration_{max}$, mainly due to *for loop* line # 12 and *while loop* line # 11 of algorithm 1 The running times presented in these tables are encouraging. After many tests, it has been noted that a swarm of 70 particles and a process of 70 iterations are largely sufficient to have good results. Moreover, most PSO-based approaches seek the best particle among the whole swarm, a new variant is to subdivide the swarm to sub-swarms, and perform the different calculations independently for each sub-swarm. The set of sub-swarms act in parallel using threads. At the end of the process, the final solution is computed using all the sub-swarms solutions. Therefore, even if the swarm size and the algorithm iterations number grow, the running time of the algorithm can easily decrease.

Table 2 Average running time of NDT-PSO *wrt* the number of iterations of the optimization process ($iteration_{\max}$) during one NDT-PSO iteration, (for $N_p = 70$, tests carried out in Fig. 7)

Number of iterations ($iteration_{\max}$)	Running time (s)
10	0.0369800877
20	0.0579088445
30	0.0718727725
40	0.0883877902
50	0.1078476976
60	0.1250863987
70	0.1457493524
80	0.1607982934
90	0.1786244514
100	0.1990917095

3.4 Comparison with SLAM methods

To evaluate the performances of the NDT-PSO algorithm with respect to other SLAM methods, it has been compared with two widely used methods in the ROS community, namely gmapping [46] and Hector-SLAM [47]. Gmapping method is based on a Rao-Blackwellized particle filter [46]. The robot's trajectory and the environment map are estimated using a particle filter, which incrementally processes the laser data (observations) and the odometric data as they are available. Hector-SLAM is based on a scan-matching approach using map gradients approximations and a multi-resolution grid [47]. The environment is represented using an occupancy grid map [62]. The tests have been carried out in the CDTA hall scenario (see Fig. 8a) in the same conditions and constraints for the three methods. The results are depicted in Fig. 8, where, it is clear that gmapping and Hector-SLAM methods (respectively in Fig. 8b, c) are not as accurate as NDT-PSO method (see Fig. 8d). From a map building perspective, both Hector-SLAM and gmapping are based

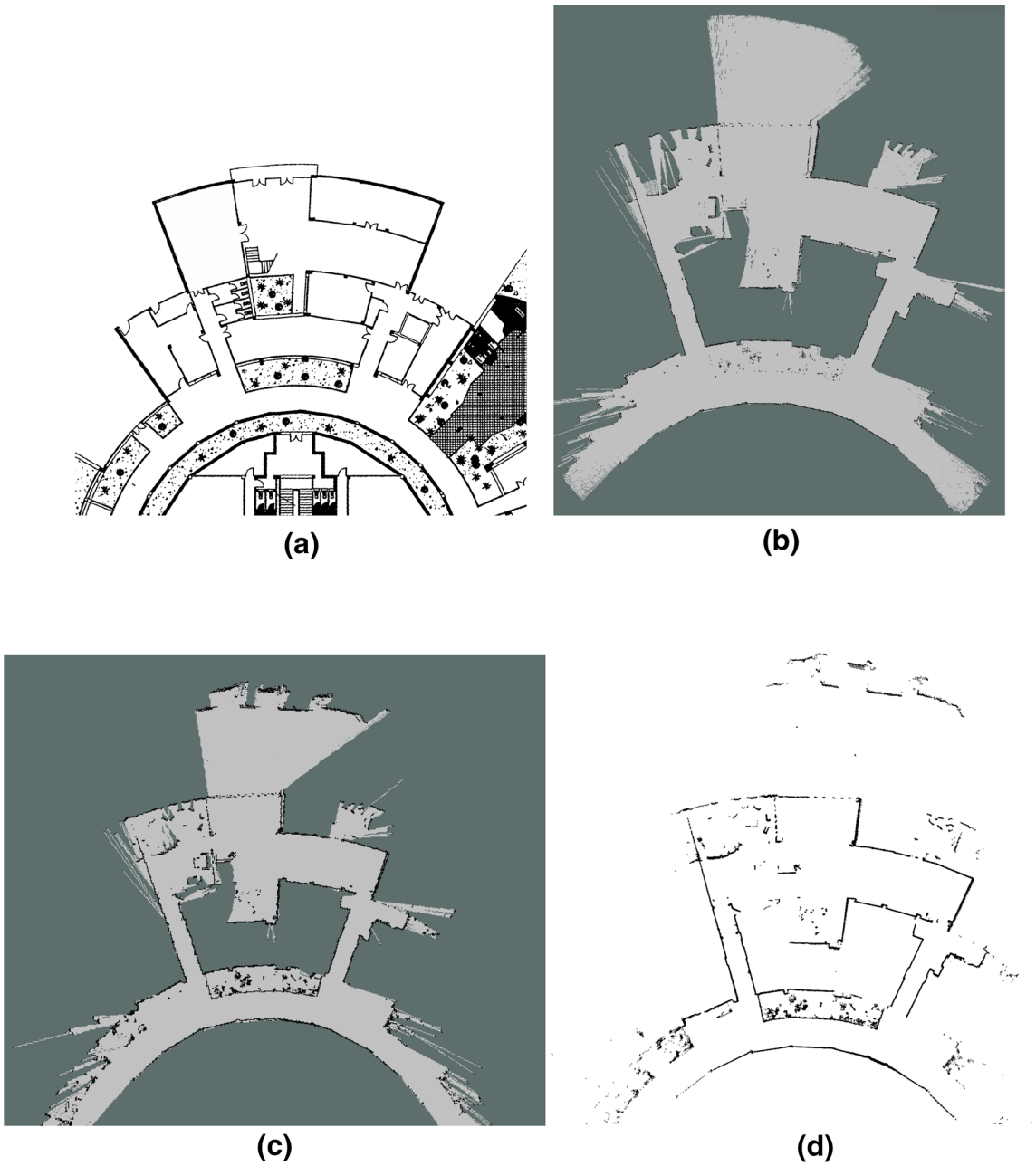

Fig. 8 Comparing NDT-PSO with gmapping and Hector-SLAM methods; **a** test scenario, **b** gmapping, **c** Hector-SLAM and **d** NDT-PSO

on a grid map representation [48], which is known to be less accurate than NDT-based representation (as stated in a previous work [41]). Moreover, to solve scan-matching problem, Hector-SLAM uses a gradient-based approach which can be subject to local minima. However, for NDT-PSO, based on PSO, a bio-inspired stochastic optimization approach presents better performances to find a global optimum [63]. Regarding gmapping, the main problem of the Rao-Blackwellized approach is its complexity in terms of number and variety of particles needed to build an accurate map. It should be also noted that the accuracy of gmapping results is invariant through different runs of the package (while maintaining the same conditions), where the presented results are the best. It can be explained by the fact that for the same input, a Rao-Blackwellized particle filter can produce different outputs. For all this reasons NDT-PSO provides better performances.

4 Conclusions

In this paper, NDT-PSO a bio-inspired stochastic approach has been proposed. It is based on particle swarm optimization algorithm to solve the pose estimation problem, which is a key process in scan-matching-based SLAM methods. The solution is encoded as the best particle in the swarm representing the best transformation between two successive NDT maps. The obtained results demonstrate the performances of NDT-PSO in real situations in both indoor and outdoor environments, either static or dynamic. The resulting map and estimated positions remain accurate even in loop closing situations and scenarios crowded with moving objects. Furthermore, it has been demonstrated that the algorithm converges rapidly and it is able to find the best particle while avoiding local minima. NDT-PSO is tested for different swarm sizes and different numbers of optimization process iterations, and the results show that a swarm of 70 particles and a process of 70 iterations are more than enough to have good results. It is also very suitable for real time applications , where the average running time of NDT-PSO for 70 particles and 70 optimization process iterations, is $145ms$. The running time varies depending on these two parameters. Moreover, NDT-PSO has been evaluated against other commonly used SLAM methods and has shown better results in term of accuracy.

As future work, this algorithm should be tested in very large scale environments for an intelligent transportation application. It could be also interesting to improve the mapping based on data fusion techniques using onboard sensors.

References

1. Dissanayake G, Durrant-Whyte H, Bailey T (2000) A computationally efficient solution to the simultaneous localisation and map building (slam) problem. In: IEEE international conference on robotics and automation (ICRA), pp 1009–1014
2. Hahnel D, Burgard W, Fox D, Thrun S (2003) An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 206–211
3. Murphy KP (1999) Bayesian map learning in dynamic environments. Adv Neural Inf Process Syst 12:1015–1021
4. Khairuddin AR, Talib MS, Haron H (2015) Review on simultaneous localization and mapping (slam). In: 2015 IEEE international conference on control system, computing and engineering (ICCSCE), pp 85-90
5. Singandhupe A, La H (2019) A review of slam techniques and security in autonomous driving. In: 2019 third IEEE international conference on robotic computing (IRC), pp 602-607
6. Cadena C, Carlone L, Carrillo H et al (2016) Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. In: IEEE Transactions on robotics, pp 1309–1332
7. Sung-Hyeon J, Ung-Hee L, Tae-Yong K et al (2018) A robust slam algorithm using hybrid map approach. In: 2018 international conference on electronics, information, and communication (ICEIC)
8. Choi J, Maurer M (2014) Hybrid map-based slam with rao-blackwellized particle filters. In: 17th international conference on information fusion (FUSION), pp 1-6
9. Zhang T, Wu K, Song J et al (2017) Convergence and consistency analysis for a 3-dinvariant-ekf slam. In: IEEE robotics and automation letters
10. Lee H, Chun J, Jeon K et al (2018) Efficient ekf-slam algorithm based on measurement clustering and real data simulations. In: 2018 IEEE 88th vehicular technology conference (VTC-Fall), pp 1-5
11. Li J, Zhong R, Hu Q, Ai M (2016) Feature-based laser scan matching and its application for indoor mapping. Sensors 16(8):1265
12. Wang D, Xue J, Tao Z et al (2018) Accurate mix-norm-based scan matching. In: IEEE/RSJ international conference on intelligent robots and systems (IROS)
13. Wang J, Fujimoto Y (2017) Combination of the icp and the pso for 3d-slam. In: 43rd annual conference of the ieee industrial electronics society, IECON
14. Zahran S, Moussa A, Sesay A et al (2018) Enhancement of real-time scan matching for uav indoor navigation using vehicle model. ISPRS Ann Photogramm Remote Sens Spat Inf Sci. https://doi.org/10.5194/isprs-annals
15. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng 137:106040
16. Namadchian A, Ramezani M, Razmjooy N (2016) A new metaheuristic algorithm for optimization based on variance reduction of guassian distribution. Majlesi J Ectr Eng 10(4):49
17. Ahmadianfar I, Bozorg-Haddad O, Chu X (2020) Gradient-based optimizer: a new metaheuristic optimization algorithm. Inf Sci 540:131–159
18. Razmjooy N, Estrela VV, Loschi HJ, Fanfan W (2019) A comprehensive survey of new meta-heuristic algorithms. Recent Advances in Hybrid Metaheuristics for Data Clustering. Wiley Publishing, New Jersey
19. Nedjah N, de Oliveira PJA (2020) Simultaneous localization and mapping using Swarm intelligence based methods. Exp Syst Appl 159:113547
20. Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng 8(1):22–34
21. Jinran Wu et al (2020) An improved firefly algorithm for global continuous optimization problems. Exp Syst Appl 149:113340
22. Razmjooy N, Ramezani M (2014) An improved quantum evolutionary algorithm based on invasive weed optimization. Indian J Sci Res 4(2):413–422
23. Razmjooy N, Khalilpour M, Ramezani M (2016) A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: theory and its application in PID designing for AVR system. J Control Autom Ectr Syst 27(4):419–440
24. Wang D, Tan D, Liu L (2018) Particle swarm optimization algorithm: an overview. Soft Comput 22(2):387–408
25. Bouraine S, Azouaoui O (2020) Safe motion planning based on a new encoding technique for tree expansion using particle swarm optimization. Robotica, pp 1-43, (in Press), Available online 10 September
26. Zhu Q, Yuan M, Liu Y et al (2014) Research and application on fractional-order darwinian pso based adaptive extended kalman filtering algorithm. Int J Robot Autom 3:245–251
27. Lee H C, Park SK, Choi J S et al (2009) PSO-FastSlam: An improved FastSlam framework using particle swarm

optimization.In: Proceedings of the 2009 IEEE international conference on systems, man, and cybernetics
28. Liu D, Liu G, Yu M (2012) An improved FastSLAM framework based on particle swarm optimization and unscented particle filter. J Comput Inf Syst 8(7):2859–2866
29. Wu S, Li P, Zhao F et al (2017) FastSlam method based on gaussian particle swarm optimization.In: Advances in social science, education and humanities research (ASSEHR), volume 130, 2nd international forum on management, education and information technology application (IFMEITA 2017)
30. Biber P, Strasser W (1996) The normal distributions transform: a new approach to laser scan matching.In: IEEE/RSJ international conference on intelligent robots and systems (IROS)
31. Stoyanov T, Magnusson M, Andreasson H et al (2012) Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. Int J Robot Res 31(12):1377–1393
32. Hong H, H. Lee B (2017) Probabilistic normal distributions transform representation for accurate 3-d point cloud registration. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3333-3338
33. Zaganidis A, Magnusson M, Duckett T, Cielniak G (2017) Semantic assisted 3-d normal distributions transform for scan registration in environments with limited structure. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4064–4069
34. Einhorn E, Gross HM (2015) Generic NDT mapping in dynamic environments and its application for lifelong SLAM. Robot Auton Syst 69:28–39
35. Li Q, Xiong R, Vidal-Calleja T (2017) A GMM based uncertainty model for point clouds registration. Robot Auton Syst 91:349–362
36. Wolcott RW, Eustice RM (2017) Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving. Int J Robot Res 36(3):292–319
37. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. IEEE World Congr Comput Intell. https://doi.org/10.1109/ICEC.1998.699146
38. Schmiedel T, Einhorn E, Gross HM (2015) Iron: a fast interest point descriptor for robust ndt-map matching and its application to robot localization. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3144–3151
39. Magnusson M, Duckett T (2015) A comparison of 3d registration algorithms for autonomous underground mining vehicles. In: Proceedings of the European conference on mobile robotics (ECMR), pp 86–91
40. Stoyanov T, Magnusson M, Almqvist H, Lilienthal AJ (2011) On the accuracy of the 3d normal distributions transform as a tool for spatial representation. In: Proceedings of the IEEE international conference on robotics and automation (ICRA), pp 4080–4085
41. Saarinen J, Andreasson H, Stoyanov T, Lilienthal AJ (2018) Normal distribution transform monte-carlo localization (ndt-mcl). In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 382-389
42. Pang S, Kent D, Cai X et al (2018) 3d scan registration based localization for autonomous vehicles—a comparison of NDT and ICP under realistic conditions. In: 2018 IEEE 88th vehicular technology conference (VTC-Fall), pp 1–5
43. Morita K, Hashimoto M, Takahashi K (2019) Point-cloud mapping and merging using mobile laser scanner. In: 2019 Third IEEE international conference on robotic computing (IRC), pp 417-418
44. Li M, Zhu H, You S, Wang L, Tang C (2018) Efficient laser-based 3D SLAM for coal mine rescue robots. IEEE Access 7:14124–14138
45. Stoyanov T, Magnusson M, Lilienthal AJ (2012) Point set registration through minimization of the l 2 distance between 3d-ndt models. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp 5196–5201:
46. Grisettiand G, Tipaldi GD, Stachniss C, Burgard W, Nardi D (2007) Fast and accurate SLAM with Rao-Blackwellized particle filters. Robot Auton Syst 55:30–38
47. Kohlbrecher S, Meyer J, Yon Stryk O et al (2011) A flexible and scalable slam system with full 3d motion estimation. In: IEEE international symposium on safety, security and rescue robotics
48. Moravec H, Elfes A (1985) High resolution maps from wide angle sonar. In: Proceedings of the IEEE international conference on robotics and automation (ICRA), pp 116–121
49. Einhorn E, Gross H M (1987) Sensor integration for robot navigation: combining sonar and stereo range data in a grid-based representation. In: 26th IEEE conference on decision and control, pp 1802-1807
50. Ypma TJ (1995) Historical development of the newton-raphson method. SIAM Rev 37:531–551
51. Olson EB (2009) Real-time correlative scan matching. In: IEEE international conference on robotics and automation (ICRA), pp 4387–4393
52. Walter E (2014) Numerical methods and optimization: a consumer guide. Springer, Berlin
53. Dor AE (2012) Improvement of particle swarm optimization algorithms: applications in image segmentation and electronics. Dissertation, University Paris-Est
54. VenkataRao R, Savsani VJ (2012) Mechanical design optimization using advanced optimization techniques. Springer, London
55. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE international conference on neural networks, pp 1942–1948
56. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73
57. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. IEEE Trans Evol Comput Swarm Intell 1(1):33–57
58. Arumugam M, Rao Senthil MVC, Chandramohan A (2008) A new and improved version of particle swarm optimization algorithm with global-local best parameters. Knowl Inf syst 16(3):331–357
59. Bonyadi MR, Michalewicz Z, Li X (2014) An analysis of the velocity updating rule of the particle swarm optimization algorithm. J Heuristics 20(4):417–452
60. Li Z, Zhu T (2015) Research on global-local optimal information ratio particle swarm optimization for vehicle scheduling problem. In: International conference on intelligent human-machine systems and cybernetics, pp 92–96
61. M'hamdi B, Teguar M, Mekhaldi A, (2016) Optimal design of Corona ring on HV composite insulator using PSO Approach with dynamic population size. IEEE Trans Dielectr Ectr Insul 23(2):1048–1057
62. Thrun S, Burgard W, Fox D (2008) Probabilistic robotics. MIT press, Cambridge
63. Jeong-Jung K, Ju-Jang L (2015) Trajectory optimization with particle swarm optimization for manipulator motion planning. In: IEEE transactions on industrial informatics, pp 620–631