NDT-PSO, a New NDT based SLAM Approach using Particle Swarm Optimization

Sara Bouraine[1], Abdelhak Bougouffa[2] and Ouahiba Azouaoui[3]

***Abstract*— This paper deals with the problem of simultaneous localization and mapping (SLAM). Providing both accurate environment's map and pose estimation is necessary to correctly navigate, which is a key issue for a mobile robot interacting with human beings. It is a line of research that is always active, offering various solutions to this issue. Nevertheless, among many SLAM methods, Normal Distributions Transform (NDT) has shown high performances, where numerous works have been published up to date and many studies demonstrate its efficiency *wrt* to other methods. In this paper a new NDT based SLAM method using Particle Swarm Optimization called NDT-PSO is proposed. The main contribution is to invest the bio-inspired approach PSO to solve pose estimation problem based on iterative NDT maps. Real experiments have been performed on a car-like mobile robot to confirm the performances of NDT-PSO approach and its efficiency in both static and dynamic environments.**

I. INTRODUCTION

Today, mobile robots are present in our daily life and they become more and more autonomous. Indeed, thanks to the ability to locate and map its environment, the robot can plan trajectories and navigate in a real world in order to perform various tasks without human intervention. This ability is known as Simultaneous Localization And Mapping (SLAM) problem [1], [2], [3]. It consists of estimating the robot position while building the map of the environment. Most developed SLAM approaches are under the Filtering [4], [5], [6] or scan matching [7], [8], [9] 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.

Indeed, in recent years, Particle Swarm Optimization (PSO) is increasingly exploited in SLAM-based methods to eliminate the problems of insufficiency and inaccuracy of priori information for Kalman filter methods [10] and impoverishment of particles for particle filter methods [11], [12], [13]. In [10], the fuzzy adaptive extended Kalman filtering method was improved by introducing the fractional-order Darwinian particle swarm optimization (PSO) to compute an accurate priori noise model. Lee et al. [11] proposed a FastSLAM framework where the robot position is estimated using Rao-Blackwellized particle filter. FastSLAM is known to degenerate over time in terms of accuracy due to the particle depletion in 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. [12] where PSO and unscented particle filter are introduced to reduce drastically the number of particles thanks to PSO for pose estimation and to improve map estimation accuracy thanks to unscented particle filter. In Wu et al. [13], to solve the degeneration of particles and the position inaccuracy (due to the need of 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 [9], the ICP and Hector SLAM algorithms are improved using the system model as an initialization step followed by the ICP or Hector as alignment step. The work in [7] introduces in the ICP algorithm both a 2D laser scan matching method based on 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. [8], a mixture of exponential power (MoEP) distributions is proposed to approximate 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 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 [14], [15], [16], [17], [18] also many studies demonstrate its efficiency *wrt* to other methods [19], [20]. However, to solve the optimization problem, most NDT-based methods use Newton Algorithm (like [21], [22]). Few works have been interested in the problem of optimization; in [19], [18], pose estimation is performed using Monte Carlo Localization (MCL). In [16], the best fitting alignment between two sets of point samples is found through the minimization of the L2 distance

[1]Sara Bouraine is with Center for development of advanced technologies (CDTA), cite 20 Aout 1956, Baba Hassen, Algiers, Algeria s_bouraine@yahoo.fr

[2]Abdelhak Bougouffa was with Center for development of advanced technologies (CDTA), cite 20 Aout 1956, Baba Hassen, Algiers, Algeria abdelhak.bougouffa@universite-paris-saclay.fr

[1]Ouahiba Azouaoui is with Center for development of advanced technologies (CDTA), cite 20 Aout 1956, Baba Hassen, Algiers, Algeria oazouaoui@cdta.dz

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 [17].

In this paper a new NDT-based SLAM method called NDT-PSO is proposed, where the scan matching process is based on PSO method. Experiments performed on a car-like mobile robot confirm the performances of NDT-PSO approach and its efficiency in both static and dynamic environments.

This paper is organized as follows: section 2 presents the proposed approach, where the scan matching problem is formalized before presenting NDT-PSO algorithm. Validations in real experiments are given in section 3. Finally, section 4 provides a summary and a conclusion of the paper.

II. PROPOSED APPROACH: NDT-PSO

To solve a SLAM problem, 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.

A. Environment Representation

In this paper, a NDT-based representation is used so as to deal with uncertainties and environment constraints. Proposed by Biber and Strasser [14], NDT is based upon a grid discretization of the space, where each 2D laser scan of the collected data is modeled as a set of cells, where a probability of measurement is associated with a 2D scan point contained in a cell.

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

- As the space is partitioned 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. $m_c = \{q_i\}, i = 1, ..., N_{m_c}$, where $m_c \subset M$.
- For each grid cell with ID c, the mean μ_c and the covariance Ω_c should be computed to determine the corresponding normal distribution. 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.

B. Pose Estimation

To solve the pose estimation problem, scan matching approaches, including NDT use gradient based approaches such as Newton method [23] with major drawbacks to be computationally expensive and sensitive to the choice of departure position, which is a frequent problem [24], [25]. In this paper, the proposed approach NDT-PSO is based on PSO, which is commonly used to solve several optimization problems due to its efficiency and simplicity and it is more likely to fall on the global minimum without position initialization [26].

Developed by Kennedy and Eberhart [27], PSO is a stochastic population-based approach inspired from an animal behaviour, namely fish schooling and bird flocking. The swarm movement is a very intelligent behaviour similar to an optimization problem where each individual is a possible solution dubbed particle, and it is formalized as follows:

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

with p a given particle, $X \in \mathbb{R}^D$ its position vector, $V \in \mathbb{R}^D$ its velocity vector, and D the search space dimension.

From equation 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 particle motion 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 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 $rand1$ 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 swarms 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 $rand2$ 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:

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

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:

$$X(p) = T = (T_x, T_y, \theta) \quad (8)$$

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:

$$q_j' = T(q_j, X) \quad (9)$$

such that:

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

The particles represent possible solutions to the scan matching problem. 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 $X(p)$. Therefore, the particles are optimized by maximizing the following objective function:

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

$P(q_j')$ 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:

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

C. NDT-PSO *Algorithm*

The overall process of NDT-PSO approach for a given iteration k is depicted in algorithm 1. Based on two successive scans M_{k-1} and M_k and given a swarm with N_p particles, 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 equations 1 and 2 (thanks to the function BUILD_MAP). It results a local normal distributions based map denoted MAP_{k-1} (line # 1, algorithm 1), which is used to compute the transformation $Tran$. Next, the swarm is initialized for the first optimization process iteration ($\tau = 0$). During this step, the whole particles are randomly initialized according to the function RANDOM_INITIALIZATION, where each particle is defined by its X_τ and V_τ vectors. Based on $X_{\tau=0}$, the personnel best $Pbest_\tau$ is initialized and the set of the 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 points set $q_j \in M_k$ into the coordinate frame of scan M_{k-1} according to equation 10 (line #37, algorithm 1). (2) Compute the normal distribution P of each mapped point q_j' according to equation 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 $Gbest_\tau$, $X_{\tau+1}$ 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 $X_{\tau+1}$ (line #18, algorithm 1) and $f_{\tau+1}$ is computed based on updated parameters. Given the maximization criterion, $Pbest_{\tau+1}$ and $Gbest_{\tau+1}$ are computed. This process is repeated for a given iteration number $iteration_{max}$. At the end, the best solution corresponding to $Gbest$ is assigned to $Tran$, and correspondingly, GMAP is updated with M'_k, such that $M'_k = \{q_j' \in M'_k / q_j' = T(q_j, Tran) \wedge Tran = Gbest\}$.

III. 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. ROS Computation graph of the NDT-PSO algorithm is depicted in Fig. 1.

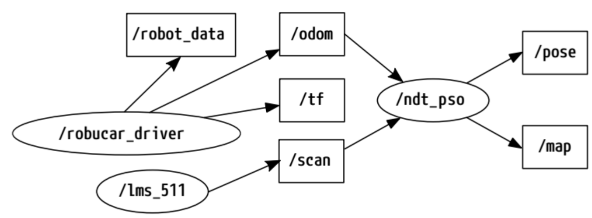


Fig. 1: ROS Computation graph generated by rqt_graph. ndt_pso node communicates with /laser and /odom topics to get sensors data and with /pose and /map topics to publish robot position and constructed map.

A. *Experimental platform*

It is a standard car-like vehicle called *Robucar*, with two fixed rear wheels and two orientable front wheels (see Fig. 2). 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.

B. NDT-PSO *at work*

NDT-PSO algorithm has been tested in two scenarios: (1) *the CDTA-hall-scenario* that shows NDT-PSO performances in indoor structured environment, and (2) *the CDTA-urban-scenario*, which is an outdoor unstructured environment. 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 $w_c = w_s = 2$ given refs. [28], [27]. The particles are randomly initialized

[1] www.ros.org

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.

MAP$_{k-1}$ $\leftarrow$ BUILD_MAP(M_{k-1}); //Map construction given equations 1 and 2
//Particles initialization (optimization iteration $\tau = 0$)
for each particle p=1 to N_p **do**
  $(X_\tau(p), V_\tau(p)) \leftarrow$ RANDOM_INITIALIZATION();
  $Pbest_\tau(p) = X_\tau(p)$;
  MAPPING_SCAN(M_k,$X_\tau(p)$);
  compute $f_\tau(p)$; // the objective function according to equation 11
end for
$Gbest_{index} = \underset{p=1,\dots,N_p}{\text{argmax}} f_\tau(p)$; //index of the particle corresponding to the global best
$Gbest_\tau = X_\tau(Gbest_{index})$;
while $\tau < iteration_{max}$ **do**
  for each particle p=1 to N_p **do**
    $V_{\tau+1}(p) = V_\tau(p) + w_c|rand_1|(Pbest_\tau(p) - X_\tau(p)) + w_s|rand_2|(Gbest_\tau - X_\tau(p))$;
    if $|V_{\tau+1}(p)| > V_{pmax}$ **then**
      $V_{\tau+1}(p) = V_{pmax}$;
    end if
    $X_{\tau+1}(p) = X_\tau(p) + V_{\tau+1}(p)$;
    MAPPING_SCAN(M_k,$X_{\tau+1}(p)$);
    compute $f_{\tau+1}(p)$; // the objective function according to equation 11
    if $f_{\tau+1}(p) > f_\tau(p)$ **then**
      $Pbest_{\tau+1}(p) = X_{\tau+1}(p)$; //update personal best
    end if
  end for
  //update global best
  if $\max_{p=1,\dots,N_p} f_{\tau+1}(p) > \max_{p=1,\dots,N_p} f_\tau(p)$ **then**
    $Gbest_{index} = \underset{p=1,\dots,N_p}{\text{argmax}} f_{\tau+1}(p)$;
    $Gbest_{\tau+1} = X_{\tau+1}(Gbest_{index})$;
  end if
end while
$Tran = Gbest_{\tau+1}$;
GMAP $\leftarrow$ UPDATE_GLOBAL_MAP($Tran$,M_k);
return $Tran$,GMAP

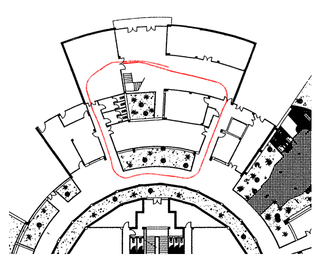
Procedure MAPPING_SCAN(M_k,X)
for each 2D point $q_j \in M_k$ **do**
  //Transform q_j into coordinate frame of the scan M_{k-1} using equation 10
  $q_j{}' = T(q_j, X)$;
  Compute $P(q_j{}')$; //The normal distribution of $q_j{}'$ according to equation 12
end for
EndProcedure

Fig. 2: Experimental platform Robucar.

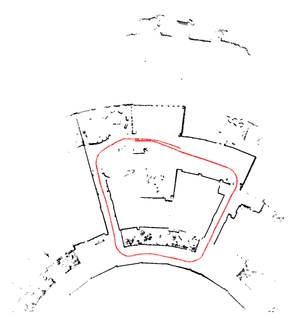
in a limited area around the previously estimated pose within a radius of $1m$ according to the cartesian coordinates and an angle of $\pi/8$ according to the orientation.

1) CDTA hall scenario: This dataset has been recorded in *the CDTA hall scenario*, in the presence of static objects of different natures (see Fig. 3a and Fig. 3b). The resulting map and robot trajectory are depicted in Fig. 3c. The map has been faithfully rebuilt and robot trajectory correctly estimated even in closing loop.

(a)



(b)



(c)

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.

2) CDTA urban scenario: The second scenario has been carried out in an urban environment at CDTA represented in Fig. 4, in the presence of arbitrary shaped objects (ex. trees, shrubs, buildings, persons, etc.). Fig. 5 illustrates the resulting map and estimated trajectory for different robot paths. These experiments show the performance of NDT-PSO in an outdoor environment. Furthermore, with only 70 particles,

Fig. 4: CDTA urban scenario.

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

NDT-PSO can find the best particle while avoiding local minima even when closing loop (see Fig. 5 at the bottom). Moreover, in the case of Fig. 6, NDT-PSO has been tested when the robot is performing three loops while maintaining a constant steering angle. The resulting robot trajectories seem perfectly superposed.

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

The NDT-based representation has already been evaluated against other SLAM methods to demonstrate its performances, like in ref. [29]. Furthermore, from an optimization perspective, PSO method is known for its fast convergence

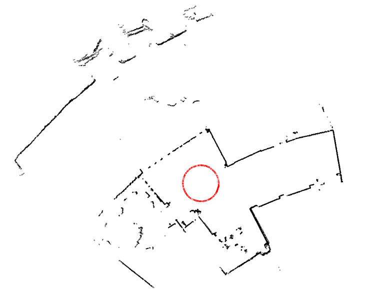


Fig. 6: NDT-PSO at work in a closing loop situation.

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

and its ability to find a global optimum [30], which justifies the above results and the effectiveness of NDT-PSO method.

C. Algorithm Performances

NDT-PSO algorithm is a function of some parameters: mainly the swarm size N_p and the number of optimization process iteration $iteration_{max}$. In the above experiments, these parameters were set to fixed values, however, to more understand their effects on NDT-PSO, the algorithm has been assessed accordingly. The tables I and II give respectively running times of NDT-PSO wrt N_p and $iteration_{max}$. From the obtained results, it can be stated 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. These running times 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.

IV. CONCLUSIONS

In this paper, NDT-PSO a bio-inspired stochastic approach has been proposed to solve a scan matching based SLAM problem. Even though this issue has been largely addressed in literature, most proposed approaches lack fast convergence and the simplicity of the optimization algorithm. The primary contribution of this paper has been to propose a new approach to solve the optimization problem using PSO method. 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

TABLE I: Average running time of NDT-PSO *wrt* the number of particles in the swarm (N_p), for $iteration_{max} = 70$ (tests carried out in Fig. 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

TABLE II: 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

indoor and outdoor environments, either static or dynamic. The resulting map and estimated positions remain accurate even in closing loop situations and scenarios crowded with moving objects. It has been also demonstrated that the algorithm converges rapidly and it is very suitable for real time applications.

This work could be extended by comparing NDT-PSO algorithm with other SLAM methods to better show its performances, which is ongoing. Furthermore, this algorithm should be tested in very large scale environments for an intelligent transportation application.

REFERENCES

[1] A. R. Khairuddin, M. S. Talib, H. Haron, Review on simultaneous localization and mapping (slam). 2015 IEEE International Conference on Control System, Computing and Engineering, pp.85–90, 2015.

[2] A. Singandhupe, H. La, A review of slam techniques and security in autonomous driving. 2019 Third IEEE International Conference on Robotic Computing (IRC), pp. 602–607, 2019.

[3] C. Cadena, L. Carlone, H. Carrillo, et al, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics, pp.1309–1332, 2016.

[4] J. Choi, M. Maurer, Hybrid map-based slam with rao-blackwellized particle filters. 17th International Conference on Information Fusion (FUSION), pp.1–6, 2014.

[5] T. Zhang, K. Wu, et al, Convergence and consistency analysis for a 3-dinvariant-ekf slam. IEEE Robotics and Automation Letters, 2017.

[6] H. Lee, J. Chun, K. Jeon, et al, Efficient ekf-slam algorithm based on measurement clustering and real data simulations. 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pp. 1–5, 2018.

[7] J. Li, R. Zhong, Q. Hu, et al, Feature-based laser scan matching and its application for indoor mapping. Sensors (Basel), 2016.

[8] D. Wang, J. Xue, Z. Tao, et al, Accurate mix-norm-based scan matching. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.

[9] S. Zahran, A. Moussa, A. Sesay, et al, Enhancement of real-time scan matching for uav indoor navigation using vehicle model. ISPRS Ann Photogramm Remote Sens Spatial Inf Sci, pp. 171–178, 2018.

[10] Q. Zhu, M. Yuan, Y. Liu, et al, Research and application on fractional-order darwinian pso based adaptive extended kalman filtering algorithm. International Journal of Robotics and Automation (IJRA), pp. 245–251, 2014.

[11] H. C. Lee, S.K. Park, J. S. Choi, et al, PSO-FastSlam: An improved FastSlam framework using particle swarm optimization. Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, 2009.

[12] D. LIU, G. LIU and M. Yu, An improved FastSlam framework based on particle swarm optimization and unscented particle filter. Journal of Computational Information Systems, pp. 2859–2866, 2012.

[13] S. Wu, P. Li, et al, FastSlam method based on gaussian particle swarm optimization. Advances in Social Science, Education and Humanities Research, volume 130, 2nd International Forum on Management, Education and Information Technology Application, 2017.

[14] P. Biber, W. Straer, The normal distributions transform: A new approach to laser scan matching. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1996.

[15] T. Stoyanov, M. Magnusson, H. Andreasson, et al, Fast and accurate scan registration through minimization of the distance between compact 3-d ndt representations. International Journal of Robotics Research, pp. 1377–1393, 2012.

[16] E. Einhorn, H. M. Gross, Generic ndt mapping in dynamic environments and its application for lifelong slam. Robotics and Autonomous Systems, pp. 28-39, 2015.

[17] R. W. Wolcott, R. M. Eustice, Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving. International Journal of Robotics Research, pp. 292-319, 2017.

[18] T. Schmiedel, E. Einhorn, H. M. Gross, Iron: A fast interest point descriptor for robust ndt-map matching and its application to robot localization. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3144–3151, 2015.

[19] J. Saarinen, H. Andreasson, T. Stoyanov and A.J Lilienthal. Normal distribution transform monte-carlo localization (ndt-mcl). IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 382-389, 2018.

[20] S. Pang, D. Kent, X. Cai, et al, 3d scan registration based localization for autonomous vehicles - a comparison of ndt and icp under realistic conditions. 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pp. 1-5, 2018.

[21] K. Morita, M. Hashimoto, K. Takahashi, Point-cloud mapping and merging using mobile laser scanner. 2019 Third IEEE International Conference on Robotic Computing (IRC), pp. 417-418, 2019.

[22] M. Li, H. Zhu, S. You, et al, Efficient laser-based 3d slam for coal mine rescue robots. IEEE Access, pp. 14124-14138, 2019.

[23] T. J. Ypma, Historical development of the newton-raphson method. SIAM review, pp. 531-551, 1995.

[24] E.B. Olson, Real-time correlative scan matching. IEEE International Conference on Robotics and Automation, pp. 4387-4393, 2009.

[25] E. Walter, Numerical methods and optimization: A consumer guide. Springer, 2014.

[26] R. VenkataRao, V.J. Savsani, Mechanical design optimization using advanced optimization techniques. Springer-Verlag London, 2012.

[27] J. Kennedy, R. Eberhart, Particle swarm optimization. IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

[28] Y. Shi and R. Eberhar, A modified particle swarm optimizer. In IEEE World Congress on Computational Intelligence, pp. 69-73, 1998.

[29] M. Magnusson, A. Nuchter, C. Lorken, A.J. Lilienthal, and J. Hertzberg, Evaluation of 3D registration reliability and speed-A comparison of ICP and NDT. In IEEE International Conference on Robotics and Automation, pp. 3907-3912, 2009.

[30] S. Bouraine and O. Azouaoui, Safe motion planning based on a new encoding technique for tree expansion using particle swarm optimization. Robotica: 1-43, in Press, Available online 10 September 2020.