

Bio-Inspired Scan Matching for Efficient Simultaneous Localization and Mapping

Nadia Nedjah[1(✉)], Luiza de Macedo Mourelle[2], and Pedro Jorge Albuquerque de Oliveira[1]

[1] Department of Electronics Engineering and Telecommunications, State University of Rio de Janeiro, Rio de Janeiro, Brazil
{nadia,pedrojorge}@eng.uerj.br

[2] Department of Systems Engineering and Computation, State University of Rio de Janeiro, Rio de Janeiro, Brazil
ldmm@eng.uerj.br

Abstract. Simultaneous Localization And Mapping (SLAM) is a fundamental problem in robotic systems. In this work, we apply a derivative free bio-inspired techniques to the scan-matching step, which is a the main step in the SLAM problem based on the exploitation of swarm intelligence. For this purpose, we have explore the swarm intelligence optimization meta-heuristics based on the firefly behavior. Aiming at reducing further the translational and rotational scan alignment errors, the proposed scan matching proceeds in two main steps, namely scan-to-scan and scan-to-map matching. The proposed firefly-based implementation of the SLAM provides a good trade-off regarding accuracy and execution efficiency.

Keywords: SLAM · Scan matching · Swarm intelligence · Firefly algorithm

1 Introduction

The Simultaneous Localization And Mapping (SLAM) problem deals with the intertwined tasks of estimating the trajectory of a moving agent and a map of the environment in which it takes place. One of the basic steps in SLAM is the scan matching.

Scan matching and the closely related image registration problem are a fundamental part of many methods developed to address the problems of simultaneous localization and mapping, structure from motion, object recognition, image fusion and 3D modeling. In most cases, the goal is to find a transformation that leads to maximum overlap between two sets of measurements taken from different viewpoints, or perhaps, from a single vantage point at two different instants in time or even between a set of measurements and a *global model* being constructed. In the context of SLAM, such a procedure is usually employed either to recover, or to enhance a gross estimate of the relative motion observed between

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12952, pp. 350–365, 2021.
https://doi.org/10.1007/978-3-030-86973-1_25

two data frames. In the specific case of laser range data, the problem of aligning measurements is often called *scan alignment* or *scan matching*. There are many different formulations of the scan-matching problem, as touching the form taken by the data and the pattern to be aligned. The most commonly encountered such variations are: scan-to-scan; scan-to-line; line-to-line; scan-to-map; map-to-map. The distinction between these formulations, and especially their reason for being, is but one of type, and level of complexity, of the pre-processing performed on the sensed data. It is thus sufficient to note that each one tries to exploit a particular characteristic of the data in order to get a better result under a specific set of conditions. For instance, the scan-to-line formulation is expected to have a better performance than the simple alignment of raw scan data, if the data in question has a strong linear trend, but may perform poorly if the entries of the covariance matrix of the scan data are of similar magnitude. The map-to-map approach tries to exploit all of the available data and is usually seen in the form of correlative matching [19]. Nonetheless, for all its advantages, like less sensitivity to outliers and more effective usage of past information, the large number of floating point operations it requires, makes it computationally expensive to perform, in some cases prohibitively so. This is said, it is of utmost importance to emphasize that the performance of the implementation of the scan-matching step is the main bottleneck for any SLAM solution. Hence, it is safe to establish that any improvement in terms of accuracy and/or execution speed regarding scan matching is most welcome in any SLAM system.

In this paper we propose a solution to the SLAM problem that is based on scan matching through Firefly bio-inspired meta-heuristic. The advantage of such an approach is apparent from the well-known multi-modality of the scan-matching problem. As far as we are concerned, this approach has not been investigated before. Furthermore, we have employed a graph-based back-end to allow this system to be consistent in exploration sessions that are either too long or cover too large an area. In both cases, a solution based exclusively on scan matching is known to fail due to the incremental nature of the estimation error.

The rest of this paper is organized in sections. First, in Sect. 2, we give a general overview of the slam problem and related challenges, including localization and mapping. After that, in Sect. 3, we describe the main steps of the firefly algorithm (FA). Subsequently, in Sect. 4, we explain the proposed method to perform robot SLAM efficiently yet accurately using the bio-inspired technique. Then, in Sect. 5, we present the performance results and their analysis. Finally, in Sect. 6, we draw some conclusions of the described works and point out some promising future directions to improve the proposed methodology.

2 The SLAM Problem

The SLAM problem consists of the concurrent estimation of the robot trajectory and a map of the environment in which a robotic entity moves in the course of its operation. A solution to this problem is required in situations in which a robot has no access to its trajectory and/or to a map of the environment wherein it

moves. Any SLAM system needs to provide a solution to the trajectory estimation, *i.e.* solve the localization problem, and to the approximate environment map building, *i.e.* solve the mapping problem. We assume that the required concepts to tackle the localization problem. We first define the necessary elements to approach the mapping problem. In Sect. 2.1, we describe the scan matching technique and in Sect. 2.2, we describe the main basic technique that have been applied later to form the basis concepts for understanding the proposed SLAM implementation, described in Sect. 4.

The concept of pose is well known and refers to a set of coordinates that are sufficient to determine the location and orientation of the robot in relation to the environment. In the case of confined systems moving in a two-dimensional $\mathbb{R}^3$ subspace, the pose must contain three coordinates. All definition of operations on poses are well defined [17].

The methods used to solve the SLAM problem that are take advantage of a pose graph are based on a formulation of the SLAM problem in terms of a graph of relationships between various frames in which the robot has met over time. These methods address the problem of state estimation as an optimization problem [9,10,13,15]. We also take advantage of the graph-based SLAM method contributed in [9].

2.1 Scan Matching

Because scan matching is at the heart of this work, we dedicate this section to explaining the idea behind such a process and its functionality as well as the different methodologies taken to approach it. It is noteworthy to point out that here we use a Light Detection And Ranging device (LiDAR). The scan-matching process aims to determine the best transformation between two point clouds obtained by some sensing instrument. Scan-matching usage, as a way of estimating the movement between two poses, from which measurements were taken, is effective as long as we can rely on the accuracy of both measurements and on the convergence of the underlying optimization process, within a reasonable time.

Existing scan-matching algorithms, although reasonably efficient, they are very sensitive to initialization. Given the advantages of applying techniques that are not only able to fully exploit the search space but also to exploit the solution space independently of function derivatives, and taking into account the fact that different heuristics present characteristics that make them better suited to solving particular types of problems, we decided that it would be advantageous and positively contributive for our work to investigate swarm optimization techniques to implement the scan-matching problem. In line with the objective of comparing the performance of swarm optimization meta-heuristics, we selected the part of the system, which is responsible for scan alignment to be implemented in such a way in order to allow the interchangeable application of any selected techniques. In this work, a map representation is used as an occupancy grid, and the scans are represented as point clouds.

Occupancy maps are a form of spatial representation based on space discretization *i.e.* associates one cell to each space region, each of which can be classified either as *full* or *empty* [17]. This type of spatial representation is based on the hypotheses of world staticity, independence between random variables that represent the map, and the fact that a cell can be completely full or completely empty. In other words, the presence of an obstacle in a part of a given space region makes one consider the region as entirely filled and what is estimated is the probability that this fact is true. The hypothesis of independence between the various cells is based on the probability distribution of the map given by the product of the probabilities of all its cells [19]. In this work, we will adopt the convention used in [19], wherein the algorithm that allows to perform cells mapping in this kinds of representation can be found.

2.2 Nearest Neighbor Search

A fundamental part of several of the steps of this work consists of searching for the nearest neighbors to a given point in a set of points. In order to give a formal definition of the problem, let $P = \{p_1, p_2, \cdots, p_N\}$ a set of points in $\mathbb{R}^2$, and a point q. The problem of searching for the nearest neighbor of point q in P is to find point $p \in P$, such that p is the closest to q. The simplest way to accomplish such a task would be to perform an exhaustive search computing the distance between any point of P to q and choosing the one that occasion the smallest distance. However, this approach is the least efficient and it is evident when the cardinality of P is large. In the proposed SLAM system, we are led to perform multiple searches for nearest neighbors, each with neighbors of entire point sets in a counterpart set, we proposed to store P in a structure of data that facilitates a fast search. The usage of KD-trees reduces significantly the time required to find a closer set of neighbors [5]. One of the most traditional algorithms this task is called Iterative Closest Point (ICP) [5]. The main advantages of ICP-based methods are their simplicity of implementation and agility when implemented using KD-Trees [3,4,8,20]. In the proposed implementation, several changes have been proposed which range from suggestions of point association heuristics to the implementation of hybrid methods that employ global optimization techniques to provide a version of ICP with the best initial estimate. Algorithm 1 presents the steps of the canonical method, whereing two sets of points A and B are aligned and T_i is a the initial pre-defined transformation matrix, w_i are the weighing coefficients for each deviation and d_{max} is the acceptable error threshold [5].

In general terms, the underlying operation can be viewed as an iteration of the following steps:

1. Matching: Associates to each pose in the source dataset its nearest neighbor in the destination set.
2. Minimization: Minimizes the error measure according to the cost function used.
3. Transformation: Transforms the poses in the source dataset using the result of the minimization process.

Algorithm 1. Steps of the canonical version of ICP

Require: N; T_i
$A = \{a_1, a_2, \cdots, a_i, \cdots, a_N\}$;
$B = \{b_1, b_2, \cdots, b_i, \cdots, b_N\}$;
$T = T_i$
repeat
  for $i = 1 \cdots N$ **do**
    $p_i = FindNearestNeighbor(a_i, T \cdot B)$
    if $\|p_i - T \cdot B\| < d_{max}$ **then**
      $w_i = 1$
    else
      $w_i = 0$
    end if
  end for
  $T = \min_T \sum_{i=1}^{N} w_i \cdot \|p_i - T \cdot b_i\|^2$
until Stopping condition
return T

3 Firefly Based Optimization

The FA is swarm intelligence method [7]. The inspiration of FA comes from the bio-luminescence of fireflies. In fact, it is precisely the periodic flickering of lights, a striking feature of these insects, which, together with the physical characteristic of the decay of light intensity, was used to elaborate this optimization strategy. In [7], it is shown that the decay of the luminous intensity I follows a relation of the type $I \propto \frac{1}{r^2}$, where r indicates the distance to the source of luminescence. The communication between social insects has been successfully used several times as an inspiration for the development of optimization techniques, such as in the cases of ants and bees [1,6].

The parameters of the FA are β_0, γ and n. The coefficient β_0 is the attractiveness of a firefly when the distance to it is zero. The constant γ determines the decay rate of the attractiveness of a firefly in terms of the distance to it. In the Algorithm 2, each firefly was denoted by s and the objective function by f.

The main operation is the combination of random motion with the attraction of each firefly by the other brighter members of the swarm. In fact, firefly s_j is attracted to the firefly s_i and its motion is defined by Eq. 1:

$$s_i{}^{t+1} = s_i{}^{t} + \beta_0 \cdot \exp(-\gamma \cdot r_{ij}^2) \cdot (s_j - s_i) + \alpha \cdot \epsilon_i, \tag{1}$$

wherein γ represents the light absorption factor, that is how light diffuses through the medium in which it propagates between one firefly and another. The parameter $\alpha \in [0, 1]$ is responsible for the random term $\alpha \cdot \epsilon_i$ of Eq. 1, and is sampled from a uniform distribution in that interval. The value ϵ_i is sampled from a Gaussian distribution. The value of r_{ij} is the distance between the fireflies s_i and s_j, as defined by the two-dimensional Euclidean distance r_{ij} between s_j and s_i

Algorithm 2. FA

Require: β_0,γ,n
1: **Initialize** the swarm
2: **repeat**
3: $\alpha^{(t)} = newAlfa()$
4: **for** $i := 1, 2, \ldots, n$ **do**
5: **Compute** $f(s_i)$
6: **end for**
7: **Sort** the fireflies s of the swarm according to their quality $f(s)$.
8: **Assign** the fittest firefly to s^*.
9: **Move** the fireflies according to Equation 1.
10: **until** Stopping criteria are met
11: **return** s^*

4 Proposed Solution

The proposed solution to the SLAM problem consists of four parts: (i) a scan matcher; (ii) a pose-graph updating system; (iii) a pose-graph optimization system; and (iv) a loop closure detection system. Algorithm 3 shows the macro-architecture of the proposed method.

Algorithm 3. Proposed SLAM Solution

1: $G := \emptyset$; $l_v := \emptyset$
2: $\mu_\lambda := 0$; $\sigma_\lambda := 0$;$\lambda := 0$
3: $\Delta P := \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$
4: $l_v(0) :=$ **Acquire_Scan**()
5: $t := 1$
6: **repeat**
7: $l_v(t) :=$ **Acquire_Scan**()
8: $\Delta P :=$ **Align_Scans**$(l_v(t), l_v(t-1))$
9: $G :=$ **Update_Graph**$(G, \Delta P, t-1, t)$
10: $G :=$ **Seek_Loop_Closure**(G, l_v)
11: $G :=$ **Optimize_Graph**(G)
12: $t := t + 1$
13: **until Halt** command is received.
14: $M :=$ **Generate_Global_Map** $(G, l_v))$
15: **return** M

4.1 The Scan Matcher

The scan matcher role, in the proposed system, is to determine the relationship that exists between pairs of LiDAR scans.

The proposed scan matcher, whose main steps are given in Algorithm 4, is designed to operate on two different, and complementary, modes. The first mode

of operation is a scan-to-scan alignment based on a point-to-point distance metric. Its operation is dependent on the minimization of the used distance metric, and it operates on a pair of scans by first determining a set of correspondences between them, based on a nearest neighbor criteria. The second mode is a scan-to-map alignment and consists of minimizing the distance between the pairs of corresponding points. The result of the aforementioned procedure is a transformation that leads to a maximal alignment between the input scans. So, the scan matcher works in two modes in an attempt to reduce uncertainties of scan matching. Hence, when the scan-to-scan mode fails to achieve an acceptable alignment (*i.e.* alignment error is above a threshold), the scan-to-map is executed, attempting to fine tune the relative position found by the scan-to-scan alignment in order to hopefully reduce the alignment error.

Algorithm 4. Align_Scans(v, v_r)

Require: t, μ_λ, σ_λ, ϵ
1: $l_c :=$ **Seek_Correspondences**(v, v_r)
2: $\left[\Delta P\ \lambda\right]^T :=$ **Minimize** Equation 2
3: **if** $\lambda > \epsilon$ **then**
4: $\quad m :=$ **Renderize_Local_Map** (v_r)
5: $\quad \left[\Delta P\ \lambda\right]^T :=$ **Minimize** Equation 4 with arguments (v, m)
6: $\quad \lambda :=$ **Calculate** the new λ using Equation 2 with ΔP as its argument.
7: **end if**
8: $\mu_\lambda := \frac{1}{t}(\lambda + (t-1)\mu_\lambda)$
9: $\sigma_\lambda := \left[\frac{t-1}{t}\left(\sigma_\lambda^2 + (\lambda - \mu_\lambda)^2\right)\right]^{\frac{1}{2}}$
10: $\epsilon := \mu_\lambda + \sigma_\lambda$
11: **return** ΔP

The objective function used for the scan-to-scan alignment, which evaluates the quadratic distance between the points of scan v and those of scan v_r in the coordinate system centered in pose ΔP, is specified by Eq. 2:

$$f(\Delta P) = \log\left(1 + \sum_{i=1}^{N} \left\| \left(\Delta P \oplus \begin{bmatrix} v_{x_i} \\ v_{y_i} \end{bmatrix}\right) - \begin{bmatrix} v_{rx_i} \\ v_{ry_i} \end{bmatrix} \right\|^2\right), \tag{2}$$

where ΔP is a relative pose and $\oplus$ is the operation of composition between a pose and a point. This composition allows us to express the points in v in the same coordinate system as the points in v_r centered at new pose ΔP. Note that the log function is computed in order to limit the wide variation of the used comparison function f. So, we impose the $\log(1+x)$ re-scale transformation [2]. Constant 1 is added to avoid the non-definition of the logarithm function on 0.

The scan-to-map alignment allows us to correct the robot pose estimation exploiting three facts: *(i)* all beams of a scan must end at obstacles; *(ii)* the robot estimated position must be empty in the current occupancy map; and *(iii)* robot must be able to reach that position in the map and continue its mapping from

thereon. The objective function for this second mode of scan-to-map matching is inspired by these two facts. The minimization of the proposed objective function attempts to find the best matching that minimizes any breach of these two facts. Moreover, the scan-to-map alignment is performed in the manner previously described via a minimization process using the swarm-based optimization meta-heuristic. In this case, the objective function takes on the form described in Eq. 3 [18]. Namely, let v_{x_i} and v_{y_i} be the Cartesian coordinates of the obstacle detected by the i-th beam of scan v and let ΔP be a relative pose. The result of the composition operation between the point (v_{x_i}, v_{y_i}) and the relative pose ΔP, in the coordinates of the local occupancy grid map, is denoted (ξ_i, ζ_i) and defined by Eq. 3:

$$\begin{bmatrix} \xi_i \\ \zeta_i \end{bmatrix} = \Delta P \oplus \begin{bmatrix} v_{x_i} \\ v_{y_i} \end{bmatrix}. \tag{3}$$

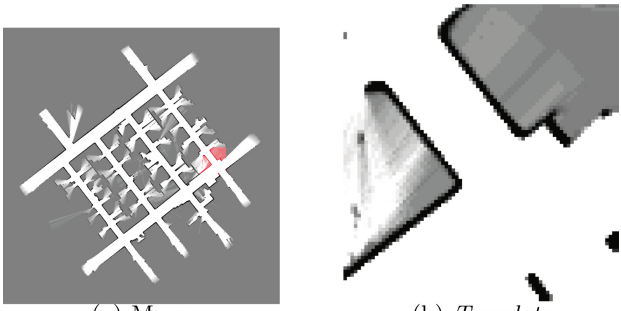
The process of scan-to-map alignment is then reduced to the minimization of the objective function g defined in Eq. 4:

$$g(\xi_i, \zeta_i) = \sum_{i=1}^{n} \frac{255 - m(\xi_i, \zeta_i)}{255} + \frac{1}{9} \sum_{k=-1}^{1} \sum_{\ell=-1}^{1} \frac{m(x_r + k, y_r + \ell)}{255}, \tag{4}$$

wherein n is the number of beams within the current scan, x_r and y_r represent the position of the robot in the local map. Variable m represents the local occupancy grid map as a matrix of unsigned eight bit integer numbers that codify the occupancy probabilities from 0 to 100% by mapping them linearly to the range from 255 to 0, respectively. So, it is established that value 0 indicates an entry in the map of 100% probability of being occupied while value 255 denotes an entry of 100% probability of being empty.

It is noteworthy to emphasize that the first summation term in Eq. 4 is with respect to all n beams within the scan one wishes to localize and thus evaluates the veracity of the first fact (*i.e.* scan beams must end at obstacles), and the second summation term is employed to penalize the presence of occupied cells in the $(\varrho + 2)^2$ adjacent cells of the robot location, taking into account the robot's size and map cell size, and thus verifying the veracity of the second fact (*i.e.* robot estimate position must be empty in the map), augmented by the extra cells around the robot location, thus guaranteeing the veracity of the third fact (*i.e.* robot must be able to reach the position and continue its trajectory from it). In both modes of operation three halting conditions have been imposed upon the optimization procedure: a limit of 1000 maximum iterations performed by each technique, a stagnation limit of 50 iterations, wherein the difference between the best and the worst solution within the swarm does not vary by more than 10^{-6}, and a tolerance threshold, whereby a solution whose corresponding objective function value is inferior to 10^{-6} is immediately accepted.

Aiming to evaluate the performance, from a mostly a qualitative point of view, of the FA meta-heuristic, we carried out an preliminary test, in which we used the map of Fig. 1 as the test-object and evaluated the mean square error as a way of check the quality of the solutions provided. Figure 1 shows the interest region while Fig. 1(b) the details of this interest region.



(a) Mapa (b) *Template*

Fig. 1. Map and template used for a preliminary evaluation of the FA meta-heuristic in the SLAM context

During the test execution, we used a limit of 1000 iterations. We performed the computational experiment with 20 fireflies with $\alpha = 0.5$, $\beta = 0.2$ and $\gamma = 1.0$, which are the default values in the code provided by the author. The number of assessments of the objective function throughout the test are shown in the histogram of Fig. 2. It presents the distribution of the number of calls to the objective function until convergence is reached.

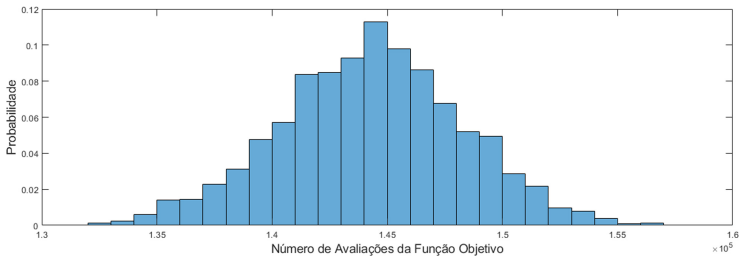


Fig. 2. Overall number of evaluation of the objective function during the test

The average number of calls observed is 1.4408×10^5, with a standard deviation of 4126.5. The overall accuracy is succinctly illustrated in the boxplot shown in Fig. 3, where it can be seen that under the imposed conditions the FA obtained a median of coordinates x, y and θ of 459.0; 547.0 and 0.0, respectively. This indicates that the FA achieved an average absolute error of 0.0% regarding coordinate x, 21.0670% regarding coordinate y and an absolute error of 0.0 radians regarding angle θ.

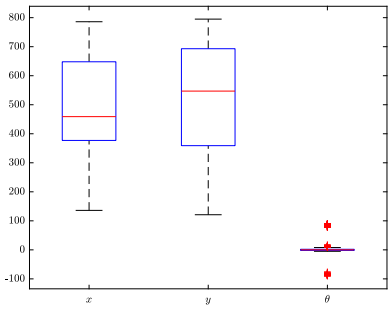


Fig. 3. *Boxplot* of the roto-translational as estimated by FA.

5 Performance Evaluation

We have performed tests using the mobile robot P3DX [16] to evaluate our system based on its accuracy and the overall speed of execution. About the accuracy, we have employed the method described in [14].

5.1 Robot Characteristics

In the performed experiments, we used the P3DX robot from Mobile Robots. The P3DX is a robot with a differential engine that weighs 9 kg, has a body composed of 1.6 mm aluminum plates, is capable of carrying a payload of up to 17 kg and has foam-filled rubber tires. The manufacturer's manual provides diagrams that specify the dimensions of the robot, one of which we reproduce in Fig. 4, where all measurements are expressed in millimeters.

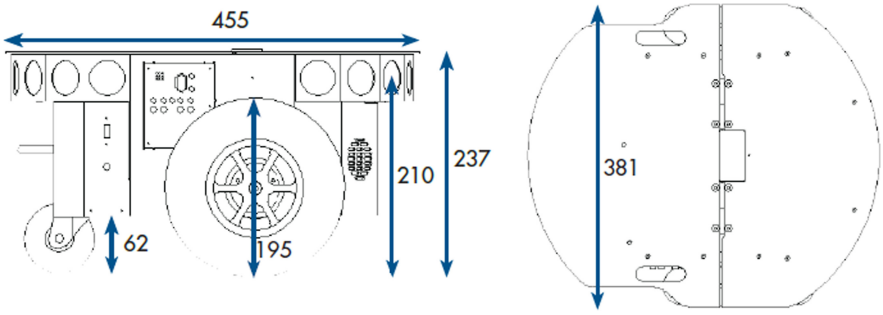


Fig. 4. Schematics with the dimensions of root P3DX.

According to the manufacturer, the robot motor is capable of reaching translational speeds of up to 1.2 m/s and angular speeds of up to 300°/s. The engine, together with all robot components and accessories, are powered by a battery

bank. The P3DX has the capacity to carry three batteries with a voltage of 12V that are capable of supplying up to 7.2Ah and whose lifespan is between eight and ten hours. The batteries are rechargeable and their recharge time is specified by the manufacturer as twelve hours. It is equipped with a SICK LMS100 laser range finder. The Laser Range Finder (LRF) has a range of 20m and sweeps a 270° circular sector in such a way that in the configuration used, 541 distance readings are returned in each sweep, that is, one measurement at each half degree.

5.2 Performance Results

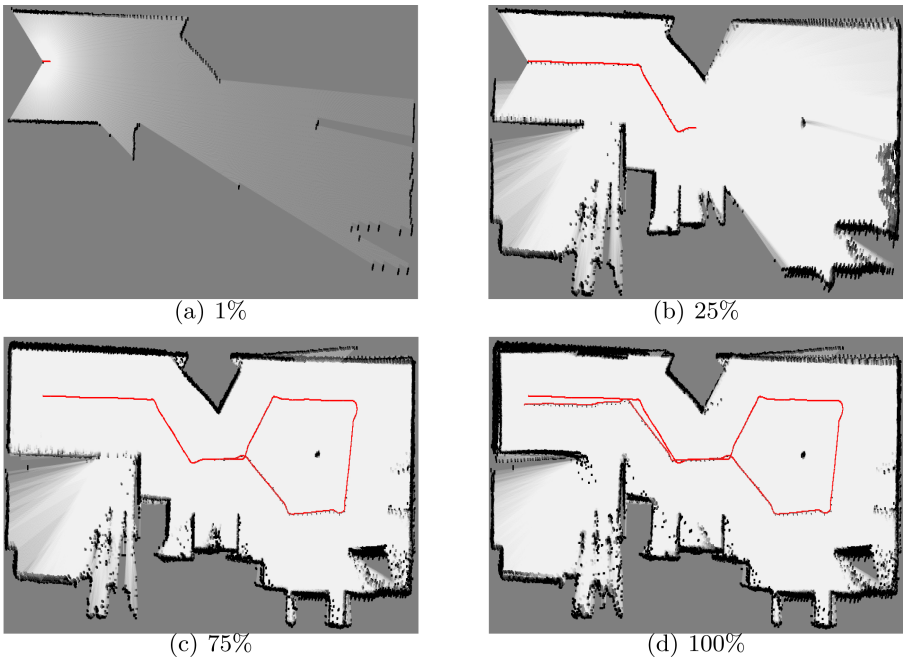
The results we obtained with robot P3DX are satisfactory, as will be shown later in this section. Indeed, even though the data collection occurred along trajectories that we had not planned, the system was able to obtain results of good accuracy in a reasonably short processing time. It is noteworthy to mention that the datasets used in the experiments, called COPA and CPSL, are of our authorship and did not have any pre-established references. This said, and with the intention of enabling an evaluation of the results presented here, we carried out an automatic and incremental alignment process combined with evaluation to build a pattern, similarly to that described in [14], which we consider the groundtruth for the obtained results.

Regarding the optimization process, the parameters of the meta-heuristics were adjusted empirically based on 30 repeated alignment tests per scenario regarding each of the used meta-heuristics. Each case was constructed based on a scan of one of the data sets that we selected for random experimentation (a random number generator was used so that the selection could take place without any human bias). We selected 10 scans and applied 3 transformation types to each scan: a pure rotation, a pure translation and a roto-translation. From the resulting 30 scenarios, two variants were generated: one without noise and one with zero mean Gaussian noise and a unit variance. Regarding the hyperparameters of FA, we set the number of fireflies $n = 20$, the randomization $\alpha = 0.75$, the attractiveness $\beta_0 = 0.2$ and the absorption as $\gamma = 1$.

The trajectory and environment where the tests take place are simple but long enough to show that the proposed system is capable of dealing with the complexity imposed by environments with diverse, and extensive, ambiguous characteristics. Through these experiments, we demonstrate the ability of the exposed system to cope, in practice with real scenarios of simultaneous mapping and localization. The location chosen for this test, although simple, demonstrates the ability to perform measurement tasks, and recognition in general, without major interventions on the part of the human operator.

In Fig. 5, we show the maps obtained by measuring the first region in a data collection stage that in a few minutes obtained 1665 measurements. The occasioned translational error is 0.75 ± 0.009 while the rotational error is 0.164 ± 0.019. Figure 6 shows some critical regions in details.

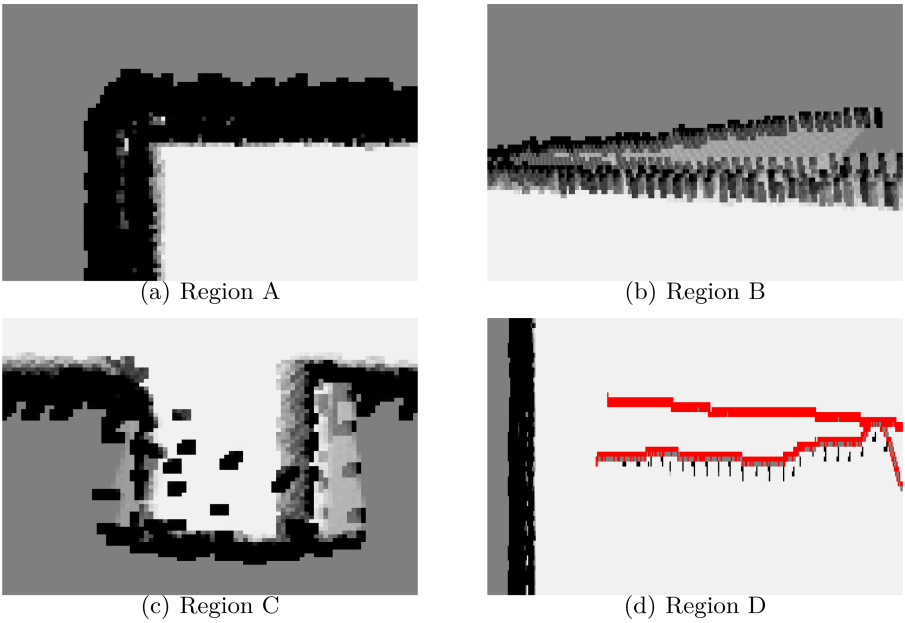
Region A presents a corner with failed alignment and the consequent replication of a section of the two walls that constitute it. This defect is observable



(a) 1% (b) 25%

(c) 75% (d) 100%

Fig. 5. The generated maps during the performed tests regarding the dataset COPA.



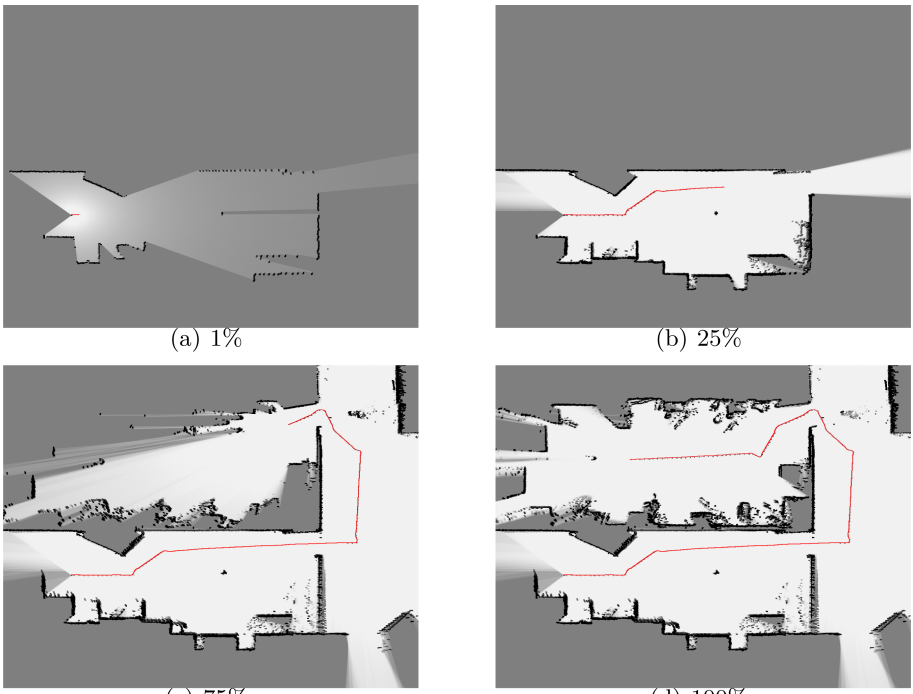
(a) Region A (b) Region B

(c) Region C (d) Region D

Fig. 6. Comparison of the four regions of interest in the built map regarding the dataset COPA.

in the built maps. The first one is shown in details in Fig. 6(a). The failure is considered less influential in this case. Region B has a stretch of wall and the presence of beams crossing solid objects. This defect is observable in the map of Figs. 6(b). Region C presents an alignment failure and the consequent replication of a stretch of wall. This defect is observable in the map of Fig. 6(c). Region D presents the replication of walls with great proximity between the replicas. This defect is observable in the built map as it can be seen in Figs. 6(d).

In Fig. 7, we show the maps obtained by measuring the second region in a data collection stage that in a few minutes obtained 1576 measurements. The occasioned translational error is 0.164 ± 0.082 while the rotational error is 0.193 ± 0.211. Figure 8 show some critical regions in details.

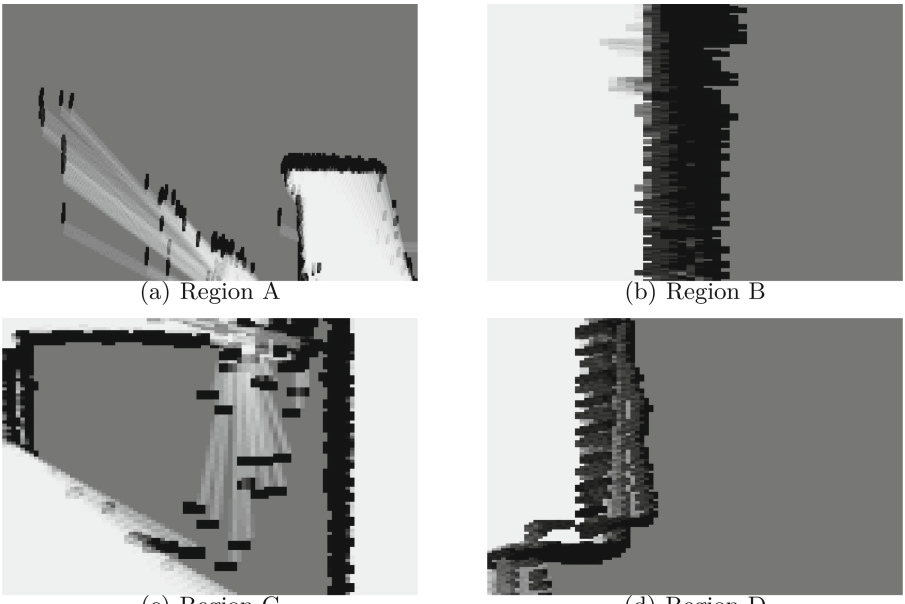


(a) 1% (b) 25%

(c) 75% (d) 100%

Fig. 7. The generated maps during the performed tests regarding the dataset CPSL

Region A presented a corner with misalignment and the consequent replication of a wall section, as shown in Fig. 8(a). Region B presents replication of a wall section and the presence of beams crossing solid objects. This defect is observable in the mapped region and shown in detail in Fig. 8(b). Region C presents beams that cross solid objects. This defect is observable in built map, as shown in Figs. 8(c). Region D presents the replication of walls and a clear imprecision in the rotational component of the poses from which the visible scans were



(a) Region A (b) Region B

(c) Region C (d) Region D

Fig. 8. Comparison of the four regions of interest in the built map regarding dataset CPSL.

performed in this region. This defect is observable in the built map, as shown in Figs. 8(d)

The results of the experiments conducted with the P3DX compared with those obtained when using Particle Swam Optimization (PSO) [12] and Artificial Bee Colony (ABC) [11] are shown in Fig. 9 and the corresponding data for the COPA and CPSL sets are presented in Table 1.

Table 1. Values of the absolute translational and rotational errors regarding the dataset COPA and CPSL.

Dataset	Type of error	ABC	FA	PSO
COPA	Translation	0.511 ± 0.164	0.164 ± 0.082	0.0721 ± 0.159
	Rotation	0.937 ± 1.124	0.193 ± 0.221	0.157 ± 0.044
CPSL	Translation	0.5 ± 0.07	0.759 ± 0.2	0.607 ± 0.159
	Rotation	0.956 ± 0.432	1.319 ± 0.279	1.043 ± 0.544

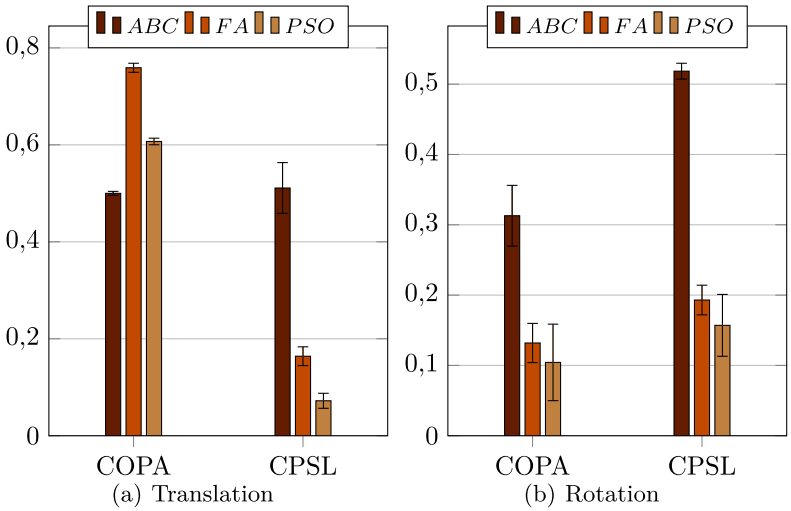


Fig. 9. Visual comparison of the absolute translational and rotational errors regarding the dataset COPA and CPSL.

6 Conclusions

In this paper, we investigated the development of a solution to the simultaneous localization and mapping problem using the swarm intelligence technique that is inspired by the firefly behaviors. The proposed scan-matching step proceeds into two main steps, aiming at further reducing the alignment error: The scan-to-scan alignment and the scan-to-map alignment. Both steps are achieved using the same swarm-based optimization meta-heuristics.

As future work, we plan to use an adaptive parameter configuration during the SLAM actual operation. Moreover, we intend to provide a parallel GPU-based implementation in order to reduce the execution time during scan alignments.

Acknowledgments. This study is financed in part by the *Coordenação de Aperfeicoiamento de Pessoal de Nível Superior*, Brasil (CAPES), Finance Code 001.

References

1. Akbari, R., Mohammadi, A., Ziarati, K.: A powerful bee swarm optimization algorithm. In: 2009 IEEE 13th International Multitopic Conference, pp. 1–6 (2009). https://doi.org/10.1109/INMIC.2009.5383155
2. Baeck, T., Fogel, D., Michalewicz, Z.: Evolutionary Computation 1: Basic Algorithms and Operators (Chapter 23). Taylor & Francis, Oxon, OX (2000)
3. Bentley, J.L.: Multidimensional divide-and-conquer. Commun. ACM **23**(4), 214–229 (1980)

4. Bentley, J.L., Friedman, J.H.: Data structures for range searching. ACM Comput. Surv. **11**(4), 397–409 (1979)
5. Besl, P., McKay, N.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
6. Dorigo, M., Caro, G.D., Gambardella, L.M.: Ant algorithms for discrete optimization. Artif. Life **5**, 137–172 (1999)
7. Fister, I., Yang, X.S., Brest, J.: A comprehensive review of firefly algorithms. Swarm Evol. Comput. **13**, 34–46 (2013)
8. Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. **3**(3), 209–226 (1977)
9. Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W.: A tutorial on graph-based SLAM. IEEE Intell. Transp. Syst. Mag. **2**(4), 31–43 (12 2010)
10. Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: incremental smoothing and mapping. IEEE Trans. Rob. **24**, 1365–1378 (2008)
11. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony ABC algorithm. J. Global Optim. **39**(3), 459–471 (2007)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE, Perth, WA, Australia, November 1995
13. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: G2O: a general framework for graph optimization. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 3607–3613. IEEE, Shanghai, China (2011)
14. Kümmerle, R.e.: On measuring the accuracy of SLAM algorithms. Autonomous Robots **27**(4), 387 (2009)
15. Lu, F., Milios, E.: Globally consistent range scan alignment for environment mapping. Auton. Robot. **4**(4), 333–349 (1997)
16. Mobile Robot: Pioneer 3dx robot (2011). https://www.generationrobots.com/media/Pioneer3DX-P3DX-RevA.pdf
17. Nedjah, N., Macedo Mourelle, L., Albuquerque de Oliveira, P.J.: Simultaneous localization and mapping using swarm intelligence based methods. Expert Syst. Appl. **159**, 113547 (2020). https://doi.org/10.1016/j.eswa.2020.113547
18. Oliveira, P.J.A., Nedjah, N., Mourelle, L.M.: SLAM baseado em scan-matching com otimização por enxame de partículas. In: Proceedings of the XIII Brazilian Congress on Computational Intelligence. Curitiba, Brazil (2017). https://doi.org/10.21528/CBIC2017-96
19. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics (Chapters 6 and 10), 1 edn. The MIT Press, Cambridge (2005)
20. Zhang, Z.: Iterative point matching for registration of free-form curves. Int. J. Comput. Vision **13**(2), 119–152 (1994)