

More Precise SLAM Using Controlled Filter Augmented with Active Perception

Luigi Maciel Ribeiro[1(✉)], Nadia Nedjah[2], and Paulo Victor R. de Carvalho[1]

[1] Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
luigimaciel@dcc.ufrj.br, paulov@ien.gov.br
[2] State University of Rio de Janeiro, Rio de Janeiro, Brazil
nadia@eng.uerj.br

Abstract. This work presents a new method, called Controlled Filter with Active Perception (CFAP), to approach the problem of Simultaneous Localization and Mapping (SLAM). SLAM aims to map an unknown environment while estimating the trajectory of a mobile agent moving within that environment. CFAP combines the two fundamental pillars of SLAM, which are scan alignment and loop closure, into a single process. The method is inspired by human perception of locating oneself and for this, it uses a mental map to guide itself. CFAP uses a Gaussian distribution to estimate possible poses and performs the alignment process in cycles, where each cycle is influenced by the results of the previous one. The Active Perception mechanism is used in each cycle to determine the quality of each of the possible poses, allowing for more accurate simultaneous localization and mapping. Benchmarking tests were performed on 5 public datasets that demonstrate the effectiveness and efficiency of the proposed method.

Keywords: Active perception · Simultaneous localization and mapping · Optimized search

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a highly complex computational problem that aims to map an unknown environment while simultaneously estimating the trajectory of a mobile agent that is moving within that environment. The fundamental pillars of SLAM are scan alignment and loop closure, which have been extensively studied in numerous works [3,5,12].

The process of scan alignment is common in our daily lives. From a young age, we are trained and encouraged to perform such processes, such as putting a key in a lock, connecting a cell phone charger, or screwing the lid onto a bottle, among others. Human alignment processes demonstrate a systematic methodology, where the process is performed in cycles, with action, observation, and adjustment stages. This concept inspired the creation of a new scan alignment

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14104, pp. 614–631, 2023.
https://doi.org/10.1007/978-3-031-37105-9_41

method, called Controlled Filter (CF), which uses a Gaussian distribution to estimate possible poses and performs the scan alignment process in stages, with the current stage being fed back with the results from the immediately preceding stage. The main inspirations of CF are Particle Filters [13] and feedback control systems [16].

The other pillar of SLAM is loop closure, which consists of the process of recognizing locations already visited and mapped by the mobile agent. In order to optimize the entire SLAM process, a scan alignment with active perception was developed, which combines the two pillars of SLAM into a single process. To demonstrate the effectiveness of the method, benchmarking tests were performed with five different public datasets.

This paper is divided into five sections: Related Works, Controlled Filter, Active Perception, Performance Results, and Conclusion. Section 2 presents works that propose optimizing the SLAM process. Section 3 demonstrates all the steps of scan alignment using the Controlled Filter. Then, Sect. 4 explains how perception is implemented to improve the SLAM process. Section 5 highlights the entire benchmarking test methodology and concludes with the test results. The paper is concluded with Sect. 6, where the obtained results are analyzed and future work is proposed.

2 Related Works

In [18], an approach to active SLAM in dynamic environments using micro aerial vehicles (MAVs) is presented. In this work, perception is used by a sensitive path planner called the Next-Best-View planner (NBVP) to avoid static and dynamic obstacles in unknown environments. Together with an active loop closure planner (ALCP), it is possible to reduce the uncertainty of SLAM and improve location accuracy. The method is demonstrated in simulated environments and presents results that confirm the effectiveness of this methodology.

In [4], the authors present a visual SLAM (VSLAM) approach with the main objective of tracking and preventing possible failures during the navigation and mapping process. Through continuous identification and association of map points during movement, this proposal can structure a planning that takes into account sensorial restrictions, resulting in reliable navigation. Perception is used in this method to predict the number of map points associated with a given pose, allowing for planning an ideal distance path so that the number of associated points is always above the limit. The results of experiments conducted in two environments demonstrate the performance of this approach.

In [8], an integrated navigation algorithm with visual SLAM is presented with the goal of optimizing the mapping and localization process. This algorithm, called perception-driven navigation (PDN), automatically balances the processes of exploration and revisiting of the environment. To make decisions, the PDN uses a reward structure that takes into account the identification of good areas for loop closure, the uncertainty in the vehicle location, and the quality of area coverage. The method is demonstrated in an autonomous application

for inspecting underwater ship hulls, using a hybrid simulation that combines synthetic and real images. The presented results confirm the applicability of the proposed method.

In [17], a method for addressing SLAM in off-road environments is proposed, which presents unique challenges that indoor SLAM does not have. These challenges include issues with direct sunlight, obstruction by foliage, uneven terrain, and chaotic and dynamic environments. To overcome these challenges, the article proposes a panoramic vision-based SLAM method that relies on the collaboration of multiple cameras and a stereo perception system to optimize localization accuracy in off-road environments. The application of this method enabled unmanned ground vehicles (UGVs) to navigate safely and reliably in dynamic and complex off-road environments. Stereo perception is used to improve vehicle motion estimation accuracy and 3D environment reconstruction, allowing for metric scale recovery. The proposed approach offers an effective and reliable solution to the problem of SLAM in off-road environments, with potential for application in a wide range of applications, from autonomous navigation of ground vehicles to mapping natural environments.

In [14], the authors present a new approach to actionable spatial perception using 3D Dynamic Scene Graphs. These graphs allow for characterizing entities in the scene and their relationships, including actionable information for planning and decision-making. The main contribution of the study is the Automatic and Integrated Spatial Perception (SPIN), which constructs a DSG from visual-inertial data. The study also describes algorithms to obtain hierarchical representations of indoor environments and their connections, as well as demonstrating the proposed spatial perception engine in a photo-realistic Unity-based simulator. This method can have a significant impact on various robotics applications, such as planning and decision-making, human-robot interaction, long-term autonomy, and scene prediction. The application of object and human detection techniques, pose estimation, and robust node inference in crowded scenes make this method even more promising for advancing robotics.

In [12], an innovative and promising approach is presented to solve the problem of simultaneous localization and mapping (SLAM) in autonomous robotics. Using derivative-free bio-inspired techniques, the research focuses on scan matching, which is a crucial step for the success of SLAM. Three swarm intelligence optimization methods were investigated: particle swarm optimization, artificial bee colony, and firefly algorithm. The proposed strategy is divided into two main stages, with the aim of reducing scan alignment errors, and uses a posture graph-based approach to maintain estimation consistency. This work was one of the main sources of inspiration for the development of the Controlled Filter with Active Perception method, which will be presented in this article. The main difference between the works is the use of active perception in the new method, which can lead to significant improvements in the accuracy and efficiency of SLAM in autonomous robotics.

3 Controlled Filter

The alignment process compares two scans, one that will be aligned and another that serves as a reference. The goal is to find a pose (x, y, ϕ) that minimizes the mean squared error between the two scans, where x is the robot's horizontal position, y is the robot's vertical position, and ϕ is the robot's angular orientation. The Controlled Filter (CF) updates one parameter at a time, making the process more efficient. For example, if the algorithm has to analyze 50 particles to find the best pose, approximately 125,000 objective function evaluations would be required. However, if the analysis is done separately for each degree of freedom (x, y, ϕ), only 150 objective function evaluations are needed per cycle. The results presented later show that the CF converges in few cycles, and using the adopted strategy, it would take more than 800 cycles to execute the objective function 125,000 times. This illustrates the higher efficiency of the CF compared to other approaches.

The CF consists of five stages: initialization, ϕ adjustment, x adjustment, y adjustment, and stopping criterion. Algorithm 1 presents the complete functioning of the filter. The CF algorithm has some input parameters, such as the bestPose, which is the pose believed to have good alignment; the currentScan, which is the measurements captured by the robot in the current pose; and the referenceScan, which is the measurements captured by the robot in the previous pose.

Algorithm 1. Controlled Filter

Require: bestPose, currentScan, referenceScan
  controlPose ← bestPose
  standardDeviation ← 0.1
  sampleSize ← 100
  maxIterations ← 100
  maxStagnation ← 3
  stagnation ← 0
  for $k \leftarrow 1$ `to` *maxIterations* **do**
    sΦ, sX, sY ← **Initialization**(bestPose, standardDeviation, sampleSize)
    bestPose ← **Adjust**Φ(sΦ, bestPose, currentScan, referenceScan)
    bestPose ← **AdjustX**(sX, bestPose, currentScan, referenceScan)
    bestPose ← **AdjustY**(sY, bestPose, currentScan, referenceScan)
    finish, stagnation, controlPose ← **stopCriterion**(controlPose, bestPose, stagnation, maxStagnation)
    if $finish = Yes$ **then**
      k ← maxIterations ▷ Finish
    end if
  end for
  return bestPose

3.1 Initialization

In this stage, the definition of the pose, believed to have a good alignment (bestPose), is performed, which can be the reference pose or a pose calculated from odometry in the first alignment cycle. After this definition, a Gaussian distribution is performed around the bestPose, considering the three degrees of freedom of the pose (x, y, ϕ). The Algorithm 2 illustrates the initialization step, which has three input parameters: the bestPose, which is the initial pose; the standardDeviation, which controls the spread of the Gaussian distribution of samples; and the sampleSize, which is the number of poses generated by the Gaussian distribution and used by the CF.

Algorithm 2. Initialization

Require: bestPose, standardDeviation, sampleSize
  sΦ ← **gaussianDistribution**(bestPose[ϕ], standardDeviation, sampleSize) ▷ Equation 1
  sX ← **gaussianDistribution**(bestPose[x], standardDeviation, sampleSize)
  sY ← **gaussianDistribution**(bestPose[y], standardDeviation, sampleSize)
  return sΦ, sX, sY

Figure 1 shows the initial data structure, where the blue points represent the reference scan and the red points represent the current scan. In Fig. 2, we present the initial pose distribution, where the red arrow represents the bestPose, and the black arrows represent the poses generated from a Gaussian distribution, where σ and μ are real values greater than zero, with σ being the mean of the distribution (bestPose) and μ being the standard deviation, according to the Eq. 1:

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{1}$$

3.2 Coordinate Adjustment

To find the best ϕ, it is necessary to keep the other parameters fixed (bestPose[x] and bestPose[y]). Then, for each ϕ, a new scan and evaluation with the objective function are performed. The objective function used is the mean squared error between the reference scan and the new scan, defined by Eq. 2:

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2. \tag{2}$$

After obtaining the best result, the bestPose[ϕ] is updated and we move on to the next step. To update the value of X, it is necessary to keep the other parameters fixed (bestPose[y] and bestPose[ϕ]). Then, for each value of X, the

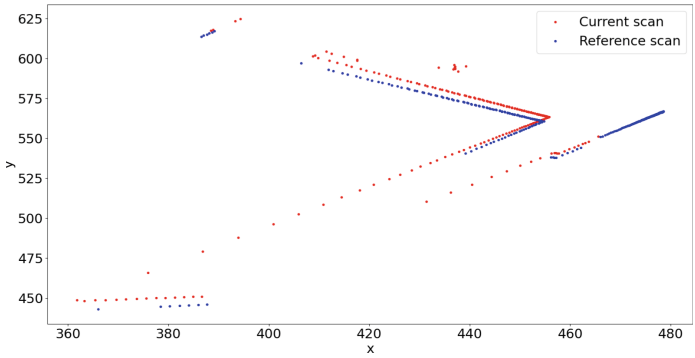

Fig. 1. Data Structure in the Initialization Step.

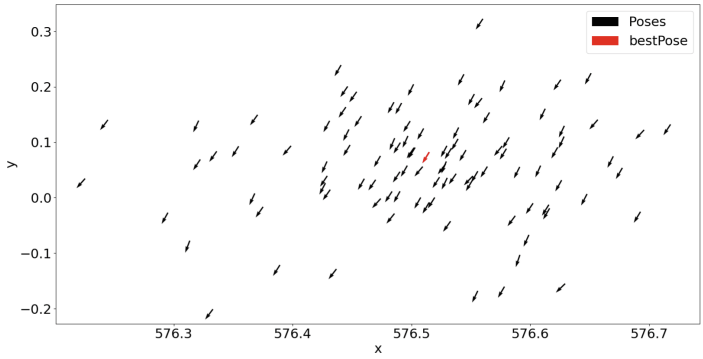

Fig. 2. Initial distribution of poses.

scan is recalculated and an evaluation with the objective function is performed. After obtaining the best result, the bestPose[x] is updated and we move on to the next step. To update the value of Y, it is necessary to keep the other parameters fixed (bestPose[x] and bestPose[ϕ]). Then, for each value of Y, the scan is recalculated and an evaluation with the objective function is performed. After obtaining the best result, the bestPose[y] is updated and we move on to the next step.

3.3 Stop Criterion

This step controls the moment of termination of the CF alignment process by checking whether there was any update in the best pose. If there was an update, the process returns to the first initialization step. If there was no update, the stagnation counter is incremented. Then, it is verified whether the maximum number of stagnation has been reached. If it has not been reached yet, the process returns to the initialization step, otherwise, the process is finished. Algorithm 3 describes the functioning of the stopping criterion and has four input

parameters: controlPose, bestPose, stagnation, and maxStagnationNumber. It returns the finalize parameter (Yes or No) and updates the stagnation counter and the controlPose parameter.

Algorithm 3. stopCriterion

```
Require: controlPose, bestPose, stagnation, maxStagnationNumber
  if controlPose = bestPose then
      stagnation ← stagnation + 1
      if stagnation > maxStagnationNumber then
          finish ← Yes
      else
          finish ← No
      end if
  else
      controlPose ← bestPose
      stagnation ← 0
      finish ← No
  end if
  return finish, stagnation, controlPose
```

4 Active Perception

Localization, as well as the different alignment processes, are common processes in our daily lives, as mentioned in the introduction of this work. We localize ourselves all the time, whether we are going to work, going from the bedroom to the bathroom, or walking around a mall. To locate ourselves, we use two things: our perception of the space around us - usually from our vision, but it can also be from other senses such as touch, hearing, or even smell - and a mental map that we build, if it is a new location, or a map that is already in our memory [15].

The active perception developed in this work is inspired by our perception when we locate ourselves. The robot, when navigating in an unknown environment, uses the map it has already built to locate itself. With this, the alignment processes of scan matching and loop closure are combined. By aligning the current pose scan to the existing map, the process solves the problem of error accumulation common in traditional alignment processes [2,10] and simultaneously identifies previously visited locations. SLAM can adopt different strategies, such as not adding new readings in already mapped areas, avoiding the inclusion of noise in the map.

SLAM follows the same methodology presented in Sect. 3, but instead of using the *MeanSquaredError* objective function, it uses the new *ActivePerception* function. This function applies a scale of punishments to determine the alignment quality, presented in Fig. 3, which ranges from 100% to 6.25%. The *ActivePerception* function evaluates each point in the scan, and for each empty point within its neighborhood, increments a punishment according to the scale of punishments. Generally, a scan point that is far away from any occupied position receives a maximum punishment, while an occupied point does not receive any punishment.

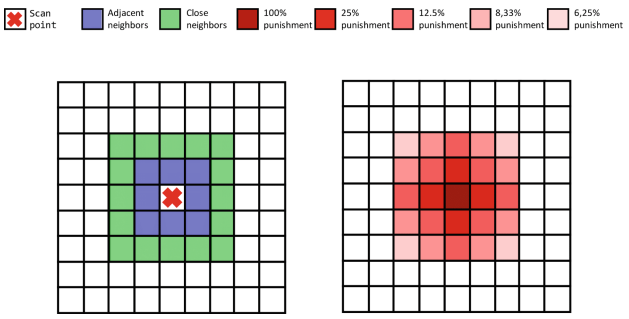

Fig. 3. Representation of the neighborhood of a scan point (on the left) and its respective penalty scale (on the right).

The function *ActivePerception* is presented in Algorithm 4. This function will be used as the objective function for the scan alignment processes performed by the Controlled Filter. As input parameters, the function receives the *adjustedPose*, which represents the pose to be evaluated, the current scan, and the map. It is worth noting that in the first SLAM cycle there is no previous map, therefore, alignment is not required. In this scenario, the map is built based on the first scan performed by the robot. From this point on, the alignment is performed normally, using the map built during previous scans.

The first process performed by the function is the conversion of the scan into coordinates. The scan is a representation of obstacle distances relative to the robot, and this representation needs to be converted to a coordinate representation, compatible with the map, in order to perform an analysis. The *adjustedPose* is defined as $P = (x_p, y_p, \phi_p)$, the coordinate as $c_i = [x_i, y_i]$, where x and y represent a position on the map, and the scan as $V = [v_1, v_2 ... v_i]$, where v_i is defined as $v_i = (d_i, \theta_i)$, with d representing the distance and θ the measurement angle. The function *scanToCoordinates* is defined by Eq. 3:

$$c_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_p + d_i \cos(\phi_p + \theta_i) \\ y_p + d_i \sin(\phi_p + \theta_i) \end{bmatrix}. \tag{3}$$

After converting the scan into coordinates, a point-by-point check is performed. For this, the *checkPoint* function is used, analyzing a specific point on

Algorithm 4. ActivePerception

Require: adjustedPose, currentScan, map
  coordinates ← **scanToCoordinates**(adjustedPose, currentScan)
  error ← 0
  penalty ← 4
  numberOfPenalties ← 0
  for *point* `in` *coordinates* **do**
    occupiedPoint ← **checkPoint**(point, map) ▷ Equation 3
    if $occupiedPoint = Yes$ **then**
      continue ▷ Move to next point
    else
      error ← error + penalty
      numberOfPenalties ← numberOfPenalties + 1
      for $neighborX \leftarrow -2$ to 2 **do**
        for $neighborY \leftarrow -2$ to 2 **do**
          neighbor ← $(point[x] + neighborX, point[y] + neighborY)$
          occupiedNeighbor ← **checkPoint**(neighbor, map)
          if $occupiedNeighbor = Yes$ **then**
            neighborDistance ← $|neighborX| + |neighborY|$
            error ← error + $\frac{penalty}{4} \frac{1}{neighborDistance}$
          end if
        end for
      end for
    end if
  end for
  return error/numberOfPenalties

the map and returning *Yes* if the point is occupied or *No* if the point is empty. The first check occurs at the point being analyzed. If the coordinate of the point is occupied, the process moves on to the next point without incrementing any penalty. Otherwise, a 100% penalty is incremented, and the analysis of all neighbors is initiated. For each unoccupied neighbor coordinate, a penalty is incremented according to the penalty scale presented in Fig. 3.

An example of the *ActivePerception* function's operation in the scan alignment process is presented in Fig. 4. Initially, a poor alignment is presented, as shown in the first figure. This alignment has several penalty points, as can be seen in the second figure. A better alignment is presented the third figure, with consequently fewer penalty points, as shown in the last figure. The goal of the alignment process is to minimize penalty points, which only occurs with an efficient alignment of the current scan to the map.

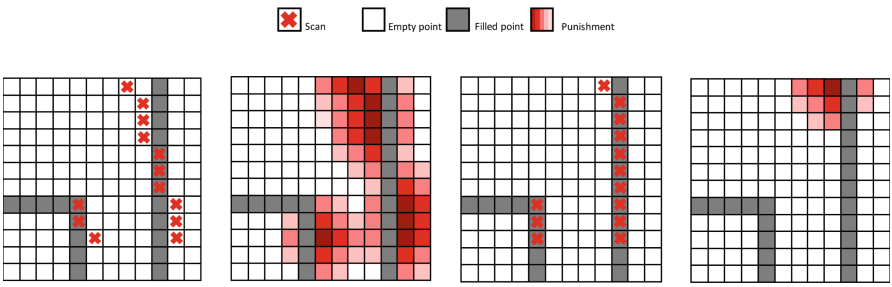

Fig. 4. Scan alignment using *ActivePerception*.

5 Performance Results

In this section, we present the implementation details of the proposed method in this work, Controlled Filter with Active Perception (CFAP). Initially, we conducted an evaluation of the filter, comparing it with a well-established search optimization method. Then, we evaluated the performance of CFAP, aiming to show the results obtained in benchmarking tests and prove its effectiveness. For this, we compared the results with other works that used the same benchmarking tests. The implementation was performed in Python, using the Pandas library for data analysis and conversion, and the PyGame library to create simulation environments. The simulations were executed on a computer equipped with an Intel Core i7-10750H processor and 8 GB of RAM memory.

5.1 Filter Performance Evaluation

The CF is a new method of search optimization, and to justify its use, we performed a comparison with another already established optimization method that has been demonstrating good results, the Particle Swarm Optimization (PSO) [1,7,11]. We compared the results of the CF with the results of the PSO, directing the analysis towards the quality of the alignment. For this, we adjusted the parameters of the CF and PSO so that the convergence time was similar. The dataset used for evaluation was the MIT CSAIL (http://www.ipb.uni-bonn.de/datasets/). Table 1 present the configuration parameters for CF and PSO, respectively. Figure 5 shows the mean squared error of 100 consecutive alignments of CF and PSO, with statistical data in Table 2. Figure 6 shows the map generated by CF and PSO, respectively, on the left and right.

Based on the presented results, the use of the new optimization search method is justified. In relation to this data sample, the CF proved to be more efficient, converging with a lower mean squared error in all alignments. In addition, the CF showed significantly lower variance and standard deviation, which represents greater stability and consistency in the results obtained.

As discussed earlier, the Controlled Filter performs scan alignment in specific degrees of freedom adjustment steps, following a predetermined order, such as

Table 1. CF (left) and PSO (right) configuration parameters.

Parameter	Value
Number of samples	100
Standard Deviation	0.1
Maximum Stagnation	3
Maximum Cycles	100

Parameter	Value
Number of particles	100
Inertia coefficient	0.8
Cognitive/Social coefficient	1.1
Maximum stagnation	10
Maximum number of iterations	1000

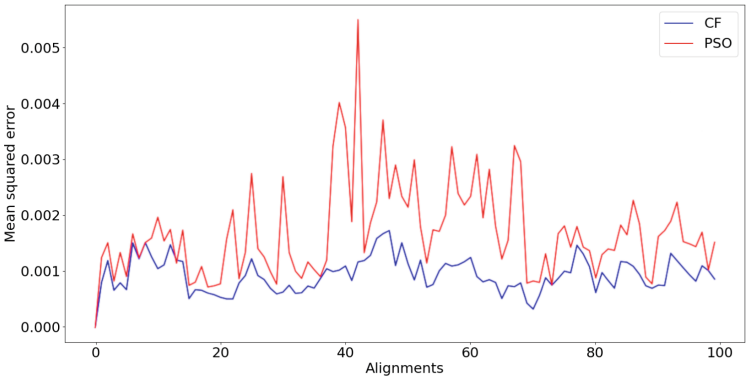

Fig. 5. Mean squared error of 100 consecutive alignments of CF and PSO.

Table 2. Statistical data of the mean squared error of 100 consecutive alignments of CF and PSO

	Mean	Variance	Standard Deviation	Maximum	Minimum
CF	0.002973	9.312295e−07	0.000965	0.005493	1.023563e−04
PSO	0.005394	7.130803e−06	0.002670	0.017439	1.845952e−04

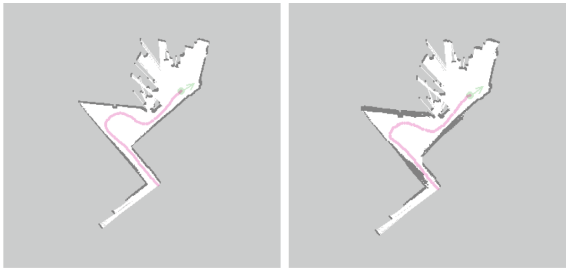

Fig. 6. Map generated by the CF (on the left) and by the PSO (on the right).

adjustment in ϕ, followed by adjustment in x and lastly in y. To verify if the adjustment order impacts the final results, it is necessary to evaluate all possible order variations. For this, an experiment was performed where only the adjustment order was varied, keeping all other simulation parameters constant, such as the scan sample, the objective function ($MeanSquaredError$), and the Controlled Filter settings. 100 consecutive scans of the MIT CSAIL dataset (http://www.ipb.uni-bonn.de/datasets/) were used for evaluation. Table 1 presents the configuration parameters of the CF, while Fig. 7 presents the results of the different possible adjustment orders in Boxplot format, defining $adjustmentOrder = [(\phi, x, y), (y, \phi, x), (x, y, \phi), (\phi, y, x), (x, \phi, y), (y, x, \phi)]$. Table 3 presents the simulation results.

Analyzing the results, it can be concluded that the variation of the adjustment order does not significantly impact the final result of the Controlled Filter. All adjustment orders present similar medians and dispensation, as well as positive symmetry.

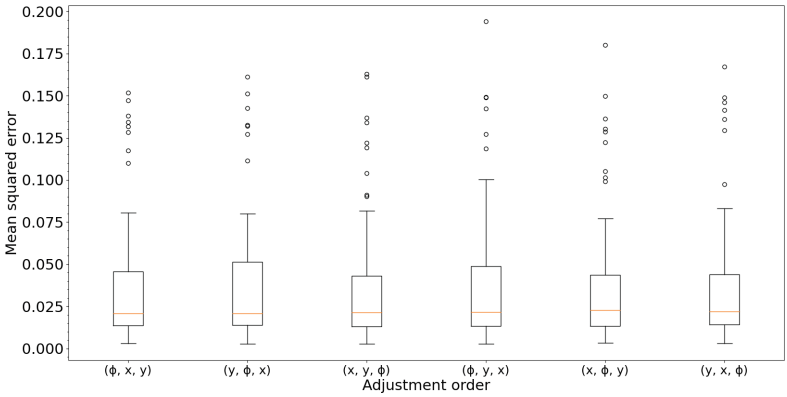

Fig. 7. Boxplots of the mean squared error of 100 consecutive alignments with different adjustment orders.

Table 3. Statistical data of Boxplots from the order of adjustment test

	1^{st}quartile	2^{nd}quartile	3^{rd}quartile	4^{th}quartile	Minimum	Outliers
(x, y, ϕ)	0.0131995	0.021235	0.0431561	0.0817251	0.0026630	9
(x, ϕ, y)	0.0133409	0.022699	0.0436485	0.0769690	0.0034185	9
(y, x, ϕ)	0.0142888	0.021827	0.0438659	0.0830524	0.0034186	7
(y, ϕ, x)	0.0140076	0.020916	0.0514501	0.0799184	0.0028686	7
(ϕ, x, y)	0.0137411	0.020817	0.0458010	0.0804070	0.0029603	8
(ϕ, y, x)	0.0133994	0.021564	0.0488546	0.1001395	0.0028788	6

5.2 CFAP Performance Evaluation

The evaluation methodology adopted was the one proposed by Kümerle in [9]. This methodology is widely applied in other works that propose solutions for SLAM, and therefore, using it makes it possible to compare the performance of the proposed method with other works in a standardized way, facilitating its evaluation. It is important to emphasize that Kümerle's methodology not only evaluates the final result of the maps but also the process of creating them, defining a model to compare the variation between consecutive poses. To evaluate accuracy, two parameters are defined: absolute translational error and absolute rotational error. The absolute translational error is defined by Eq. 4:

$$\lambda_{tranlacional} = \frac{1}{N} \sum_{i,j}^{N} |T(\delta_{i,j} - \delta^*_{i,i})|, \tag{4}$$

where N is the total number of considered poses; $\delta_{i,j}$ is the variation between poses, defined by Eq. 5:

$$\delta_{i,j} = \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_\phi \end{bmatrix} = \begin{bmatrix} (x_i - x_j)\cos\phi_j + (y_i - y_j)\sin\phi_j \\ -(x_i - x_j)\sin\phi_j + (y_i - y_j)\cos\phi_j \\ \phi_i - \phi_j \end{bmatrix}, \tag{5}$$

where $\delta^*_{i,j}$ is the variation between poses of the reference set; and $T(.)$ separates the translational component (x, y) of the relative pose and calculates the resultant using the Pythagorean theorem. The absolute rotational error, where $R(.)$ separates the rotational component (ϕ) of the relative pose, is defined by Eq. 6:

$$\lambda_{rotacional} = \frac{1}{N} \sum_{i,j}^{N} |R(\delta_{i,j} - \delta^*_{i,i})|. \tag{6}$$

With the aim of enabling comparison of the results obtained by CFAP with other works and establishing a standardized evaluation method, we chose to use public datasets available at http://www.ipb.uni-bonn.de/datasets/. In Fig. 8, it is possible to visualize the grid maps and robot trajectories corresponding to the datasets INTEL, MIT CSAIL, ACES3, FREIBURG BLDG 79 and MIT KILLIAN COURT, arranged from top to bottom, respectively.

The CFAP demonstrated success in all benchmarking tests performed, proving its ability to deal with various data structures, such as different numbers of Lidar readings, variations in measurement accuracy, and distinct topologies. This diversity is presented in the datasets used in the tests, providing notoriety to the test. The method showed a significant improvement in the generated map when compared to maps generated only with odometry. However, difficulties were identified in datasets with very long corridors and distant loop closures, due to the Active Perception methodology itself. In the case of long corridors, perception is greatly influenced by the robot's current position, which can lead to stagnation, causing the problem of corridor shortening. Regarding distant loop

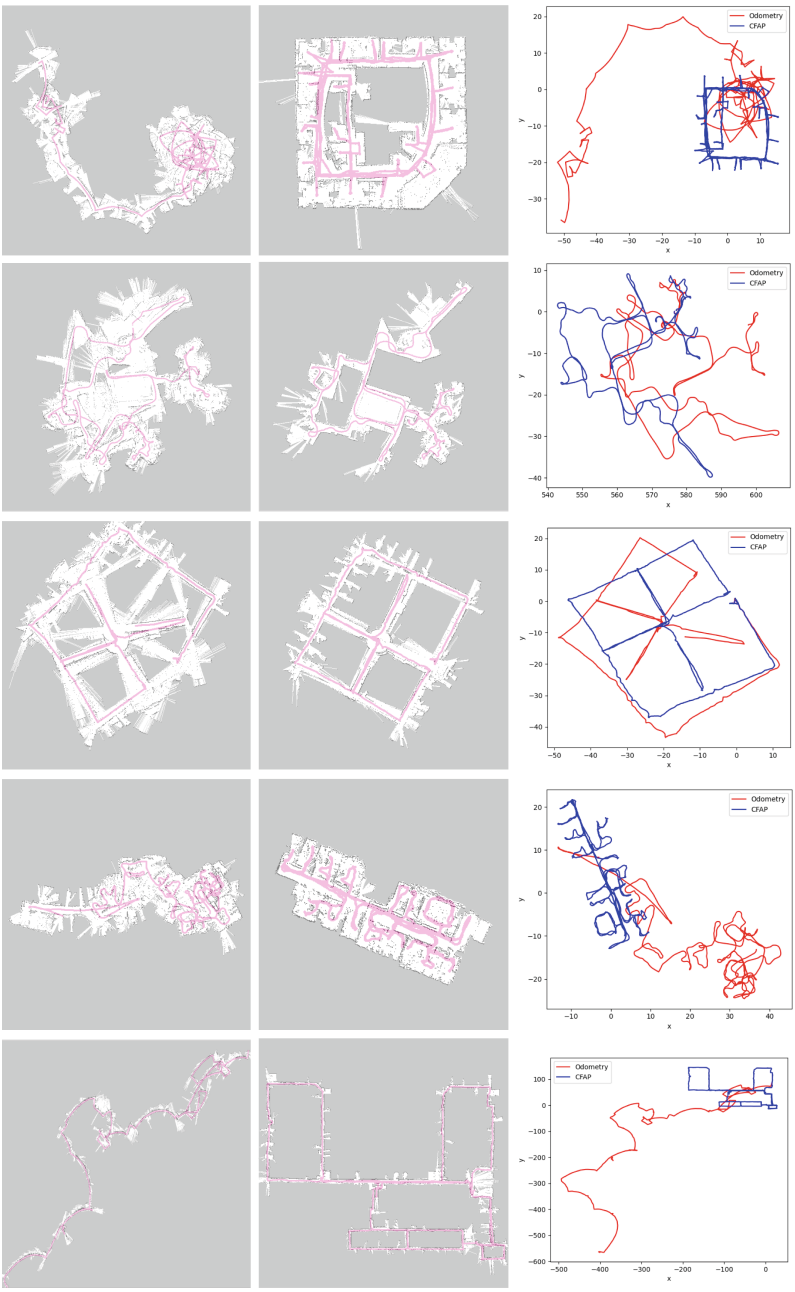

Fig. 8. Grid map and robot trajectory (pink line) obtained using only odometry (left), and using CFAP (center) and comparison of trajectories (right), simulations related to the datasets: INTEL, MIT CSAIL, ACES3, FREIBURG BLDG 79 and MIT KILLIAN COURT, from the first row to the last row, respectively. (Color figure online)

closures, perception has a limited range, which means that, in the occurrence of small alignment errors at the beginning of the loop, this error can cause a significantly large disturbance. In this scenario, Active Perception will not have the reach to correct the error.

5.3 Comparative Analysis

The comparative analysis conducted evaluates the performance of CFAP in comparison to two other popular methods for solving SLAM: Cartographer [6] and Graph Mapping (GM) [9]. The selection of datasets for this comparison was based on the recommendations provided by these studies.

Cartographer proposes a methodology for real-time loop closure with high precision. It achieves this by using branching structures and sub-maps to limit scan correspondences. Essentially, the algorithm builds a map by incrementally matching new sensor data with existing map data while maintaining loop closure. The algorithm also optimizes the map using least-squares optimization.

GM, on the other hand, is a graph-based SLAM approach. It builds a graph from sequential measurements, and then optimizes the graph configuration using a mean squared error minimization approach. Essentially, the algorithm uses a graphical representation of the environment and the robot's path, where nodes in the graph represent poses and edges represent the constraints between those poses. The optimization step then adjusts the poses to minimize the errors in the constraints.

According to the results presented in Table 4 and illustrated as a bar graph in Fig. 9, CFAP shows promising results compared to Cartographer and GM. In the INTEL dataset, CFAP outperformed both methods in terms of rotational and translational errors. Specifically, CFAP showed an improvement of 5.67% and 58.27% in translational and rotational errors, respectively, compared to Cartographer. Compared to GM, CFAP showed an improvement of 30.32% and 85.46% in translational and rotational errors, respectively.

In the MIT CSAIL dataset, CFAP still outperformed Cartographer in terms of translational and rotational errors, with improvements of 67.08% and 65.58%, respectively. However, CFAP showed a significant decrease in performance compared to GM, with an increase of 262% and 254% in translational and rotational errors, respectively. This highlights the importance of specific dataset evaluation when considering the performance of SLAM methods.

In the ACES3 dataset, CFAP showed a decrease in performance compared to Cartographer, with an increase in translational and rotational errors of 43.0% and 33.13%, respectively. Compared to GM, CFAP also showed a decrease in performance, with an increase of 30.66% and 25.65% in translational and rotational errors, respectively.

In the FREIBURG BLDG 79 dataset, FCPA showed improvements compared to Cartographer and GM in terms of rotational and translational errors. Specifically, FCPA showed an improvement of 10.61% and 13.01% in translational and rotational errors, respectively, compared to Cartographer. Compared

to GM, FCPA showed an improvement of 27.85% and 22.0% in translational and rotational errors, respectively.

Finally, in the MIT KILLIAN COURT dataset, FCPA showed a 49.93% deterioration in absolute translational error compared to Cartographer. However, there was an 8.8% improvement in absolute rotational error compared to Cartographer. Additionally, FCPA showed a 36.62% deterioration in absolute translational error and a 35.8% improvement in absolute rotational error compared to GM. These results suggest that FCPA is a promising method for SLAM and can provide greater accuracy on certain datasets when compared to commonly used methods.

Table 4. Quantitative comparison of error with [6,9]

	CFAP	Cartographer	GM
INTEL			
$\lambda_{tranlacional}$	0.0216 ± 0.0302	0.0229 ± 0.0239	0.031 ± 0.026
$\lambda_{rotacional}$	0.189 ± 1.854	0.453 ± 1.335	1.3 ± 4.7
MIT CSAIL			
$\lambda_{tranlacional}$	0.0105 ± 0.0114	0.0319 ± 0.0363	0.004 ± 0.009
$\lambda_{rotacional}$	0.127 ± 0.255	0.369 ± 0.365	0.05 ± 0.08
ACES3			
$\lambda_{tranlacional}$	0.0658 ± 0.0708	0.0375 ± 0.0426	0.044 ± 0.044
$\lambda_{rotacional}$	0.538 ± 0.365	0.373 ± 0.469	0.4 ± 0.4
FREIBURG B79			
$\lambda_{tranlacional}$	0.0404 ± 0.0258	0.0452 ± 0.0354	0.056 ± 0.042
$\lambda_{rotacional}$	0.468 ± 0.571	0.538 ± 0.718	0.6 ± 0.6
MIT K. COURT			
$\lambda_{tranlacional}$	0.0789 ± 0.0651	0.0395 ± 0.0488	0.050 ± 0.056
$\lambda_{rotacional}$	0.321 ± 0.278	0.352 ± 0.353	0.5 ± 0.5

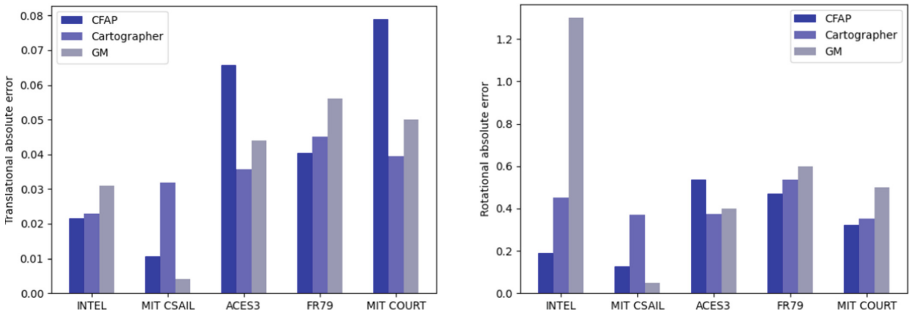

Fig. 9. Comparison graphs of absolute translational error (on the left) and absolute rotational error (on the right).

6 Conclusion

This work presented a new approach to the problem of Simultaneous Localization and Mapping (SLAM) based on scan alignment and loop closure, which are the fundamental pillars of SLAM. The systematic alignment methodology carried out by humans inspired the creation of a new scan alignment method, called Controlled Filter (CF). This method uses a Gaussian distribution to estimate possible poses and performs the scan alignment process in stages. Additionally, an active perception scan alignment was developed that combines the two pillars of SLAM into a single process.

To evaluate the proposed method, the methodology proposed by Kümerle was used, which evaluates both the final results of the maps and the process of creating them. Public data sets were used to allow for comparison of CFAP results with other works and a standardized evaluation method. The CFAP was successful in all benchmarking tests performed, but encountered difficulties in data sets with very long corridors and distant loop closures, due to the active perception methodology itself.

A comparative analysis was carried out with other SLAM methods, such as Cartographer and Graph Mapping, and CFAP obtained better results on average than the others. However, we acknowledge that there is room for improvement in CFAP, especially regarding the difficulties faced in long corridors and distant loop closures. In this sense, future work could explore the development of an auxiliary system to CFAP that detects this type of topology using perception itself, allowing for the adoption of approaches to improve the performance and accuracy of the method.

In summary, this work presented promising results and opened the way for future studies in the field of SLAM, contributing to the development of new methods and techniques that can be applied in various areas, such as robotics, mapping, and autonomous navigation.

Acknowledgments. This work is supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES - Brazil, https://www.gov.br/capes), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - Brazil, https://www.gov.br/cnpq) and by Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ - Brazil, https://www.faperj.br) with grant numbers 201.013/2022, 200.850/2021 and 211.000/2021. We are most grateful for their continuous financial support.

References

1. Amrani, R., et al.: Investigation of structural and electrical properties of ITO thin films and correlation to optical parameters extracted using novel method based on PSO algorithm. Bull. Mater. Sci. **46**(1), 8 (2023)
2. Chen, L.H., Peng, C.C.: A robust 2D-SLAM technology with environmental variation adaptability. IEEE Sens. J. **19**(23), 11475–11491 (2019)
3. Davison, A.J., Murray, D.W.: Simultaneous localization and map-building using active vision. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 865–880 (2002)

4. Deng, X., Zhang, Z., Sintov, A., Huang, J., Bretl, T.: Feature-constrained active visual slam for mobile robot navigation. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 7233–7238. IEEE (2018)
5. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006)
6. Hess, W., Kohler, D., Rapp, H., Andor, D.: Real-time loop closure in 2D LiDAR SLAM. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1271–1278. IEEE (2016)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN 1995-International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
8. Kim, A., Eustice, R.M.: Perception-driven navigation: active visual slam for robotic area coverage. In: 2013 IEEE International Conference on Robotics and Automation, pp. 3196–3203. IEEE (2013)
9. Kümmerle, R., et al.: On measuring the accuracy of slam algorithms. Auton. Robot. **27**, 387–407 (2009)
10. Mo, J., Islam, M.J., Sattar, J.: Fast direct stereo visual SLAM. IEEE Robot. Autom. Lett. **7**(2), 778–785 (2021)
11. Moazen, H., Molaei, S., Farzinvash, L., Sabaei, M.: PSO-ELPM: PSO with elite learning, enhanced parameter updating, and exponential mutation operator. Inf. Sci. **628**, 70–91 (2023)
12. Nedjah, N., de Oliveira, P.J.A., et al.: Simultaneous localization and mapping using swarm intelligence based methods. Expert Syst. Appl. **159**, 113547 (2020)
13. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House (2003)
14. Rosinol, A., Gupta, A., Abate, M., Shi, J., Carlone, L.: 3D dynamic scene graphs: actionable spatial perception with places, objects, and humans. arXiv preprint arXiv:2002.06289 (2020)
15. Thornton, I.M., Pinto, J., Shiffrar, M.: The visual perception of human locomotion. Cogn. Neuropsychol. **15**(6–8), 535–552 (1998)
16. Van de Vegte, J.: Feedback Control Systems. Prentice-Hall, Inc. (1994)
17. Yang, Y., Tang, D., Wang, D., Song, W., Wang, J., Fu, M.: Multi-camera visual slam for off-road navigation. Robot. Auton. Syst. **128**, 103505 (2020)
18. Zhao, Y., Xiong, Z., Zhou, S., Wang, J., Zhang, L., Campoy, P.: Perception-aware planning for active SLAM in dynamic environments. Remote Sens. **14**(11), 2584 (2022)