

Enhanced Monte Carlo Localization with Visual Place Recognition for Robust Robot Localization

Javier Pérez · Fernando Caballero · Luis Merino

Received: date / Accepted: date

Abstract This paper proposes extending Monte Carlo Localization methods with visual place recognition information in order to build a robust robot localization system. This system is aimed to work in crowded and non-planar scenarios, where 2D laser rangefinders may not always be enough to match the robot position within the map. Thus, visual place recognition will be used in order to obtain robot position clues that can be used to detect when the robot is lost and also to reset its positions to the right one. The paper presents experimental results based on datasets gathered with a real robot in challenging scenarios.

1 Introduction

The FROG project¹ is an FP7 research action funded by the European Commission that aims to deploy a guiding robot in touristic sites involving outdoor and partially outdoor scenarios. While robot guides have been developed since more than a decade [16], the project considers as new contributions the development of social behaviors and their adaptation by integrating social feedback, as well as the robust operation in outdoors crowded scenarios. It aims to

This paper is an extension of work presented at ICARSC 2014 [14] and is partially supported by the FP7 FROG Project (Contract 288235) funded by the European Commission and the PAIS-MultiRobot Project (TIC-7390) funded by Regional Government of Andaluca

J. Pérez and L. Merino
University Pablo de Olavide
Seville (Spain).
E-mail: jiperlar@upo.es, lmercab@upo.es

F. Caballero
University of Seville
Seville (Spain).
E-mail: fcaballero@us.es

¹ <http://www.frogrobot.eu>



Fig. 1 The FROG project aims to deploy a guiding robot with a fun personality, considering social feedback, in the Royal Alcázar of Seville and the Zoo of Lisbon. Left: Visitors surrounding the robot in the Royal Alcázar, interfering in sensor readings and interrupting robot’s trajectory. Right: FROG partners following a tour at Lisbon Zoo.

demonstrate a long-term operation of the robot in the Lisbon Zoo (Portugal) and the Royal Alcázar in Seville (Spain) (see Fig. 1).

Navigating in these crowded places (the Royal Alcázar may have more than 5000 visits per day) requires a robust localization system. Achieving long-term localization involves several issues, like handling of variant environments, error recovery, efficient place recognition, etc. Furthermore, those algorithms based on vision and visual place-recognition have to deal with illumination changes, different weather and daylight conditions. Besides that, these scenarios may present a highly variable environment with partial sensor occlusions due to the visitors, which can cause troubles to map-based localization using laser readings and dead reckoning [12].

Approaches based on 2D laser scan matching are the most extended localization algorithms for mobile robots in GPS-denied environments, due to their high accuracy compared to other sensors like ultrasonic sensors, and with a low processing cost compared to vision sensors [1]. These algorithms make use of a geometric map and scan matching to guess the new position of the robot from previous ones, and dead reckoning. Scan matching can handle small variations in the environment, such as changes of state of doors, but it can perform poorly when large variations are present, as it can be seen in crowded and dynamic environments like the Lisbon Zoo and the Royal Alcázar, where people may approach and surround the robot driven by curiosity or while they are being guided.

Several approaches have been considered to enhance the robustness of localization systems. Thus, Hentschel and Wagner [9] and Dayoub and Duckett [5] present in their works environmental representations for autonomous mobile robots that continuously adapt over time, inspired by human memory and storing the current as well as past knowledge of the environment, using sensory memory, short-term memory and long-term memory.

Online loop-detection algorithms based on scene-recognition like OpenFabMap2 [8], DLoopDetector [7], and others [2] that make use of Bag-of-Words [19] approaches have been presented to look for revisited places, what is help-



Fig. 2 Left: the FROG robot platform. It shows the main platform and the positions of the sensors. Right: Estimation of persons poses on the screen, given by stereo cameras.

ful for recovering from localization errors. Corke et al. [4] present an algorithm for getting invariant images for long-term localization based on scene appearance. They describe how to convert different time outdoor colour images to greyscale invariant ones by considering the response of the colour channels in trichromatic vision and removing illumination effect.

These visual algorithms can be easily used to provide additional localization hypotheses to the pose estimated by using other sensorial modalities, like laser rangefinders. These new hypotheses can be used to enhance the robustness of the localization system. In this paper we propose a localization algorithm based on a Monte Carlo Localization filter fed with particles from appearance clues obtained from images, which will be able to recover from possible errors in localization, combining the high accuracy of lasers with a re-localization process.

The structure of the paper is as follows: next section describes the robotic platform used for experiments. Then, Section 3 presents the map building approach. Our algorithm for robust localization is described in Section 4. The paper ends with Section 5, which presents the experimental results obtained at the Lisbon Zoo (Portugal) and the Royal Alcázar (Spain), followed by Section 6, which details conclusions and future work.

2 The robot platform

Figure 2 shows a picture of the robot considered as deployed in the Royal Alcázar for a demonstration of its capabilities. The robot platform consists of a skid-steering platform, with 4 wheels adapted to the scenarios considered in the paper. It has an autonomy of two to four hours depending on the type of ground and the number of embedded PCs running, up to three. The robot weights 80Kg approximately and its maximum velocity is 1.6 m/s (software limited to 0.8 m/s).

The robot is equipped with a wide range of sensors for safety, localization and navigation. Among them, the following sensors are considered for robot localization and navigation:

- Odometry is computed by reading encoders and angular velocities from an IMU from XSense
- Three laser rangefinders are considered. Two deployed horizontally forward and backwards, employed for localization and obstacle avoidance. The third laser is placed in the front and tilted 45 approximately in order to perceive 3D obstacles in front of the robot.
- A stereo pair, employed for person detection, 3D perception and being its left camera also used as image stream for place recognition.

3 Map building

The real experiments shown in this paper were conducted at the Lisbon Zoo and the Royal Alcazar. Both are GPS-denied areas, so a SLAM solution was chosen to create an accurate map.

The application considered in FROG allows for an offline SLAM solution: the robot can be deployed in the scenario to gather data and a map can be built offline (even though map management will be needed to add future changes in the environment). Thus, the map is obtained offline solving the full-SLAM problem, consisting on obtaining the map and full robot trajectory given all the measurements available.

The full-SLAM problem can be casted as a non-linear least-squares minimization problem, in which the sensorial data provides constraints among the different variables of the problem, typically robot poses and map feature positions. The non-linear minimization is carried out by the *SLAM back-end*. In particular, the backend employed here is *g2o* [11], which requires an initial estimation of the values of all variables, as well as the constraints between them, encoded as a graph (or hypergraph).

This graph is provided by the *SLAM front-end*. In our case, we solve the *pose-SLAM* problem, where only the trajectory of the robot is recovered by the *SLAM back-end*. Our front-end considers odometry and loop closures provided by the OpenFabMap2 [8] algorithm over the images to provide constraints on the state variables, in this case the robot poses (see Fig. 3).

After the execution of the previous minimization, an optimal corrected robot trajectory is obtained. This robot trajectory is then used to build a map from the sensor data available. For instance, a 2D or 3D map can be constructed from the laser scans and the stereo vision system. Figure 3 shows the resulting 3D map of the projected laser for a trajectory of 1.4 km. at the Lisbon Zoo. It can be seen how the odometry divergence distorts the map with respect to its real form and how the loop-closing detection allows refining the map and obtaining a globally consistent estimation.

However, while the robot trajectory is globally consistent, the simple projection of sensorial data (for instance, laser rangefinders, point clouds or stereo

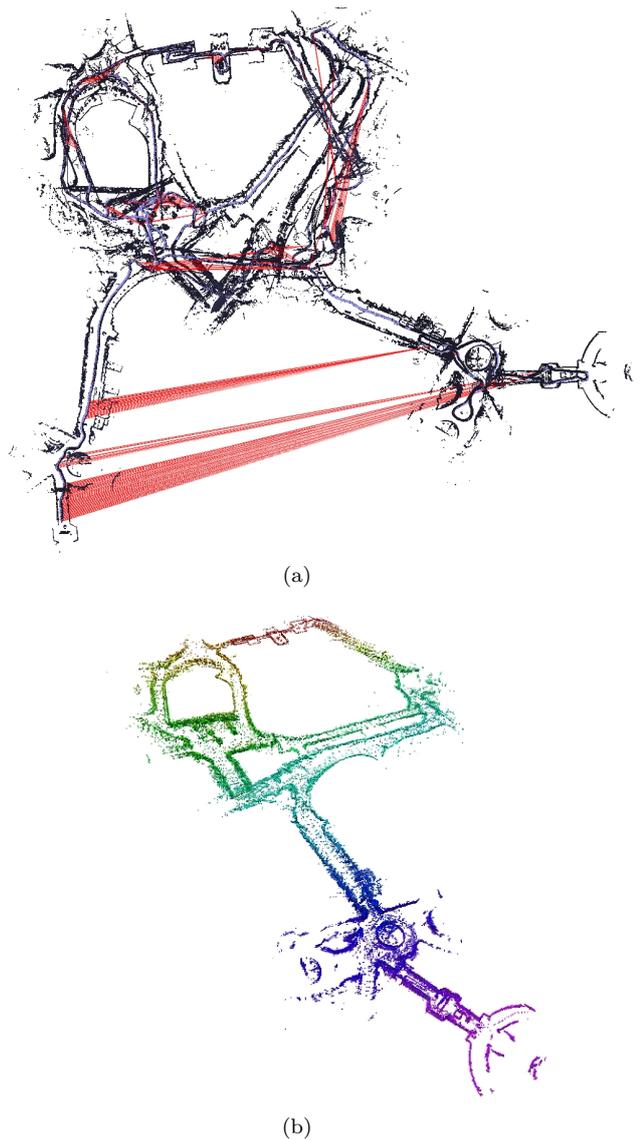


Fig. 3 (a) Loop closures obtained at the Lisbon Zoo (red lines, between revisited places). It can be also appreciated the typical drift associated to odometry. (b) Point cloud obtained from the Lisbon Zoo. The height of the points is color-coded (reddish colors indicating higher ground).

data) in the global frame will lead to maps with slight local errors, such as fuzzy walls or double walls, as the information from those sensors was not directly considered in the minimization process (see Fig. 4, top).

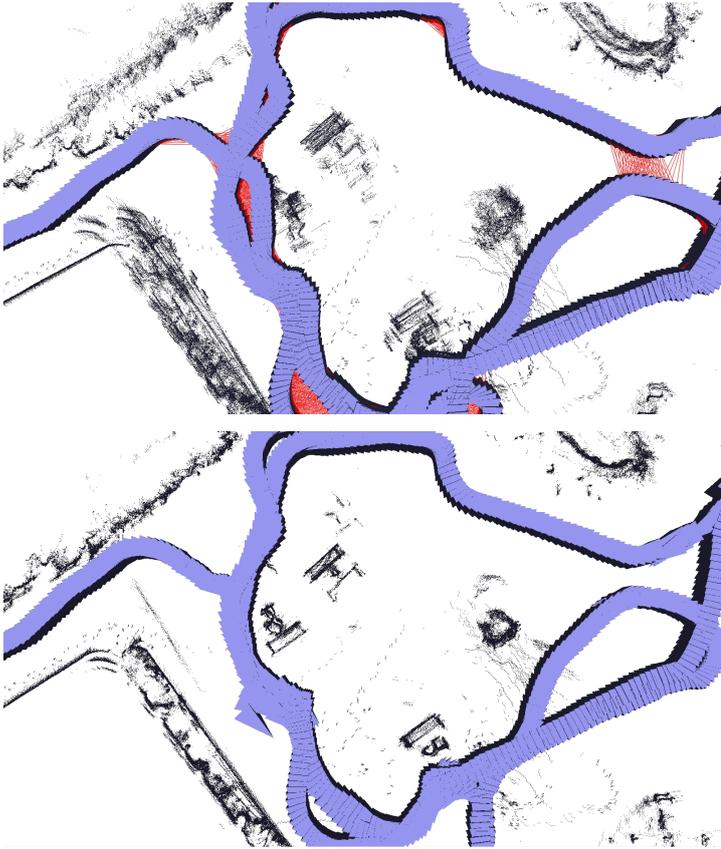


Fig. 4 Top: map obtained by projecting the information with the optimized robot pose. Red lines refer to new constraints obtained by scan matching. Bottom: Map refinement by considering the laser information into the optimization process. It can be seen how the grass on the top left is correctly aligned. Also, some trees are much better resolved.

Several approaches have appeared in the last years to cope with this issue. For instance, in [15], the authors consider a model of the laser scans based on line segments (tangents on surfaces), and include the estimation of these line segments within the optimization process, so solving a complete SLAM problem. This approach is called SSA (Sparse Surface Adjustment), in the fashion of Sparse Bundle Adjustment (SBA). This model, however, is more suitable for indoor scenarios. Recently, in [10] the same ideas as in [15] were applied, adding a robust outlier rejection within the optimization process. Nevertheless, the approach is still the same and more suitable to indoor scenarios.

As some of the scenarios in which the FROG robot has to be deployed, such as the Lisbon Zoo, present irregular structures, we have defined a different approach. In this approach, we will not include map features in the SLAM process, but will directly consider the sensorial information from the laser rangefinders to develop new constraints for the pose-SLAM problem that con-

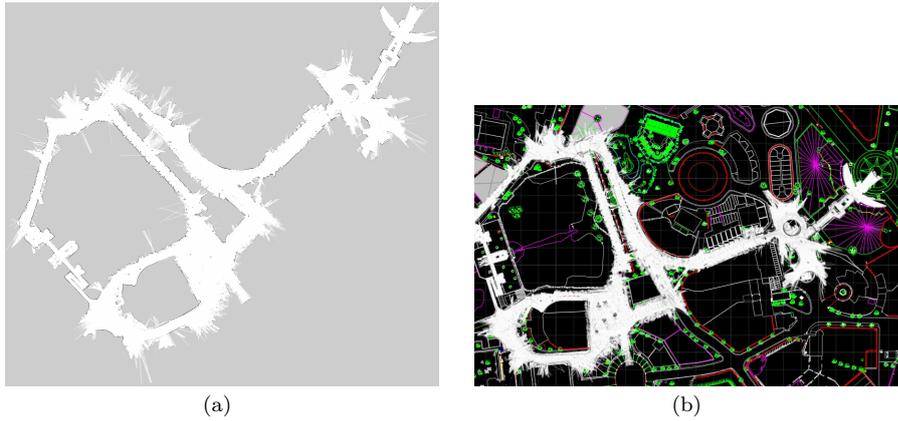


Fig. 5 (a): Resulting occupancy grid map of Lisbon Zoo after 2D laser scan integration. (b): 2D occupancy map of the Lisbon Zoo overlaid on a CAD drawing of the Zoo.

sider the errors on the map being built. Therefore, a final procedure is used to optimize the resulting map. The following steps are carried out (as published in [13]):

1. A new set of constraints is obtained by performing scan matching between pairs of laser scans or point clouds. As an initial good estimation of the poses of the robot is already available from the initial solution, the scan matching process is performed not only between consecutive robot poses, but also between close poses in space but not in time (see Fig. 4).
2. The poses are refined by minimizing an error function for these constraints which depends on the quality of the alignment of scans.
3. The initial seed for the minimization is provided by the previous solution.

In this scenario there are many surfaces that cannot be approximated by planar or smooth surfaces. Thus, the error function is directly the Iterative Closest Point scan-matching error function. At every iteration, the data association of points between scans is recomputed for the newly constrained poses, as well as the Jacobian of this error function, which is then employed by the non-linear minimization.

Figure 4 shows an example of this refinement. The final resulting map after 2D laser scan integration is shown in Fig. 5. A comparison of laser scan integration and CAD map is shown in the figure. The robot made a complete exploration of navigable area for map generation, acquiring data with both frontal and back laser. As can be seen in the Fig. 5, non-permanent obstacles like pedestrians are eliminated due to data integration when building the map (only consistent obstacles are included).

4 Robust Localization Algorithm

4.1 Base localization algorithm

The localization module should provide the robot pose in 6D to the rest of the robot modules. A map-based localization approach is employed, and therefore this pose is actually the pose with respect to the map. In particular, a Monte Carlo Localization (MCL) approach is employed [18]. Particle filters are very flexible representing arbitrary probability distributions, and allow the fusion of information coming from different sensorial inputs, which is relevant for the approach presented here.

The 6 degree-of-freedom (DOF) pose of the robot is represented by $\mathbf{x}_t = [x \ y \ z \ \gamma \ \varphi \ \theta]^T$, where we represent the orientation by the roll (γ), pitch (φ) and yaw (θ) angles. However, as we are considering a ground robot, the robot is bounded to navigate on the 2D surface of the scenarios considered. Thus, the z coordinate is actually dependent on the x and y coordinates and the map M . Furthermore, the IMU onboard the robot provides a stable solution for γ and φ by using internal filters. In order to reduce the state space required to be covered with the particles, we consider a Rao-Blackwellized filter [6], in which the current 2D pose $[x \ y \ \theta]^T$ is tracked by using a particle filter, while the height z is tracked by means of a Kalman filter (and the roll and pitch angles are provided by the IMU).

Therefore, our distribution probability on the pose of the robot is represented by a set of ω -weighted particles $\langle x_t^{[i]}, y_t^{[i]}, \theta_t^{[i]}, \bar{z}_t^{[i]}, \sigma_{z,t}^{[i]}, \gamma, \varphi, \omega^{[i]} \rangle$. These particles are updated by using the information coming from the odometry measurements (linear and angular velocities) and the laser rangefinders of the robot. A brief description of the filter can be found in Algorithm 1.

The sensors on the FROG platform allow us to obtain the motion of the robot. Wheel encoders provide the velocity on the robot frame v . Furthermore, the XSense IMU provides an estimation of the angular velocity around the robot's Z axis, $\dot{\theta}$, as well as stabilized values for roll and pitch. Given the linear and angular velocities, the particles are then propagated in line 6 by sampling from a typical probabilistic kinematic model [17].

Then, the height is then updated in lines 10 and 11 by considering the height map $h_M(x, y)$ built during the mapping phase by discretizing the XY plane and determining the height at every cell. In principle, this map may suffice to determine the height of the robot given its x and y coordinates. However, we integrate the estimation based on the odometry and that on the map in order to smooth the height estimation in case of coarse height maps.

Finally, the weights of the particles are updated by computing the likelihood of the current laser rangefinders in lines 17 and 18.

Algorithm 1 Base Localization Algorithm

```

1:  $\langle x_t^{[i]}, y_t^{[i]}, \theta_t^{[i]}, \bar{z}_t^{[i]}, \sigma_{z,t}^{[i]}, \gamma, \varphi, \omega_t^{[i]} \rangle_{L_i}^L$  Current state of the filter /* Prediction stage */
2: if Odometric measurement  $\mathbf{u}_t = [v \ \dot{\theta} \ \gamma_{imu} \ \varphi_{imu}]^T$  then
3:    $\varphi \leftarrow \varphi_{imu}$ 
4:    $\gamma \leftarrow \gamma_{imu}$ 
5:   for  $i = 1$  to  $L$  do
6:      $\langle x_{t+1}^{[i]}, y_{t+1}^{[i]}, \theta_{t+1}^{[i]} \rangle \leftarrow \text{sample\_kinematic\_model}(x_t^{[i]}, y_t^{[i]}, \theta_t^{[i]}, \mathbf{u}_t, \Delta t)$ 
7:      $\mathbf{v}_g^{[i]} = \mathbf{R}(\gamma, \varphi, \theta_t^{[i]}) [v \ 0 \ 0]^T$ 
8:      $\hat{z}_{t+1}^{[i]} = \bar{z}_t^{[i]} + \Delta t \mathbf{v}_{g,z}^{[i]}$ 
9:      $\hat{\sigma}_{z,t+1}^{[i]2} = \bar{\sigma}_{z,t}^{[i]2} + \sigma^2$ 
10:     $\bar{z}_{t+1}^{[i]} = \hat{z}_t^{[i]} - \frac{\hat{\sigma}_{z,t+1}^{[i]2}}{\hat{\sigma}_{z,t+1}^{[i]2} + \sigma_{z,M}^2} (z_t^{[i]} - h_M(x_t^{[i]}, y_t^{[i]}))$ 
11:     $\bar{\sigma}_{z,t+1}^{[i]2} = \frac{\hat{\sigma}_{z,t+1}^{[i]2} \sigma_{z,M}^2}{\hat{\sigma}_{z,t+1}^{[i]2} + \sigma_{z,M}^2}$ 
12:     $\omega_{t+1}^{[i]} = \omega_t^{[i]} \mathcal{N}(\hat{z}_{t+1}^{[i]}; h_M(x_t^{[i]}, y_t^{[i]}), \hat{\sigma}_{z,t+1}^{[i]2} + \sigma_{z,M}^2)$ 
13:  end for
14: end if
15: if Laser measurement  $\mathbf{z}_t$  then
16:   for  $i = 1$  to  $L$  do
17:    Compute likelihood  $p(\mathbf{z}_t | x_{t+1}^{[i]}, y_{t+1}^{[i]}, \theta_{t+1}^{[i]}, M)$ 
18:    Update weight  $\omega_{t+1}^{[i]} = p(\mathbf{z}_t | x_{t+1}^{[i]}, y_{t+1}^{[i]}, \theta_{t+1}^{[i]}, M) \omega_t^{[i]}$ 
19:  end for
20: end if
21: Normalize weights  $\{\omega_t^{(i)}\}, i = 1, \dots, L$ 
22: Resample if necessary

```

4.2 Appearance-based particle injection

The resampling process and occlusions of laser rangefinders may introduce errors in localization in certain executions [3], causing the particle filter to diverge or to converge to wrong locations. Intelligent re-sampling techniques can be used to limit these effects [17], but they cannot be completely avoided. For that reason we propose to extend the base algorithm in order to use information from sensors of different modalities than the laser rangefinders. In particular, the main idea is to introduce appearance information coming from the images by using the algorithm OpenFabMap2 [8]. OpenFabMap2 is a probabilistic framework for appearance based navigation and mapping using spatial and visual appearance data based on a bag-of-words approach to detect loop-closures. As OpenFabMap2 does not implement any treatment for illumination variances, it is necessary to record data at different hours or even weather seasons to improve the accuracy of matches.

Algorithm 2 summarizes the main ideas: during the mapping stage detailed in Section 3, left images from the stereo pair are gathered at regular space intervals and included into the OpenFabMap2 database tagged with their respective positions. This database is loaded and used in a modified OpenFabMap2 algorithm, that will compare the present image I_t with the stored database, which is not modified during execution. If a match between

Algorithm 2 Particle injection based on OpenFabMap2 place recognition

```

1: BoW database  $\langle x_i, y_i, \theta_i, BoW_i \rangle$ 
2: if New image  $I_t$  then
3:   Extract  $BoW_t$  from  $I_t$ 
4:   if match between present  $BoW_t$  and  $BoW_k$  from database then
5:     Evaluate error in pose
6:     if Error in position  $>$   $threshold_{position}$  OR error in orientation  $>$ 
        $threshold_{orientation}$  then
7:       Substitute  $pf\_th\%$  particles with others distributed with center pose  $\langle x_k, y_k, \theta_k \rangle$ 
       and weight  $w_{injected}$ 
8:     end if
9:   end if
10: end if

```

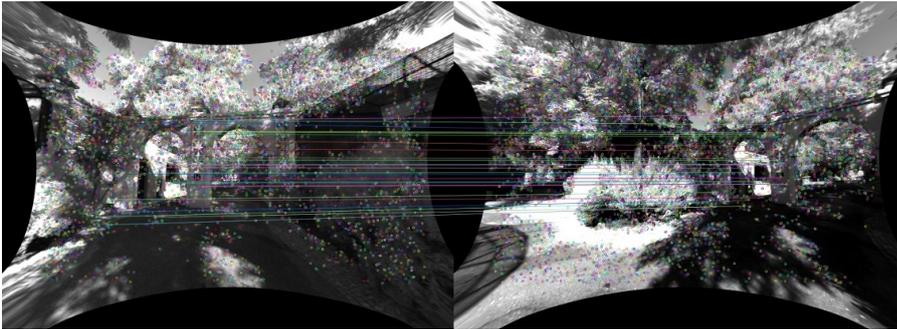


Fig. 6 An example of image matching provided by OpenFabMap2.

images I_t and image I_k of the database is detected (Fig. 6), the algorithm will evaluate the pose error between the current robot pose (according to the most-likely particle) and the stored pose of I_k . If the error is over a predefined threshold for position and orientation some particles will be injected into the current sample set.

The process of Particle Injection consists of replacing the pf_th less significant particles, where pf_th is a percentage of the current number of particles in the particle set (value of 1% in our experiments), by new particles generated from a Gaussian distribution centered at the pose where the image I_k was taken, and a new predefined weight $w_{injected}$ relative to maximum weight in the present set of particles (value of 50% in our experiments).

The Particle Injection is done in the base algorithm before evaluating the laser measurements (before line 15 in Algorithm 1), so the new inserted particles will get their weights updated according to the likelihood described by the laser measurements. After this, the algorithm will continue with the resampling process, favoring the particles with higher weights. This process of injection will not affect the particles with highest weight if the robot is well localized, allowing the localization module to have a permanent and fast recovery process from errors in localization and kidnapping problem

Table 1 Mean manually recoveries (times per test). It can be seen that MCL with Particle Injection never needed manually recovery, while plain MCL had to be recovered several times to avoid non valid divergences in position and orientation errors (see Tables 2 and 3).

Test (Max Laser Range)	MCL	MCL + Particle Injection
5m	2.5	0
10m	0.83	0
15m	0.66	0
20m	0	0

Table 2 Comparative Mean and Std. deviation for position error for both MCL and MCL+Particle Injection against ground truth pose

Max Laser Range	Position error (m.)	
	MCL	MCL + Particle Injection
5m	10.73 ± 13.64	1.95 ± 6.59
10m	10.39 ± 19.74	1.27 ± 5.14
15m	2.27 ± 6.87	1.04 ± 4.85
20m	2.83 ± 6.18	1.10 ± 6.55

Figure 8 shows the mean absolute error in position and orientation of the six trials with different laser configuration for both Particle Injection and plain MCL with respect to the ground-truth trajectory. It can be seen how the proposed approach have smaller mean errors than MCL and, more importantly, that MCL had to be manually recovered when necessary in every configuration test, while our particle injection algorithm did not need to be manually recovered. The reason of these manually recoveries is to avoid non-valid divergences in position and orientation error analysis (Tables 2 and 3). By manually recovering, both methods can still be compared in the rest of the trajectory as can be seen in Fig. 8.

It can be seen in Fig. 8 how Particle Injection gives a fast and strong recovery even in the worst scenarios of laser occlusion and people surrounding, dramatically reducing the mean number of times the robot get lost down to zero. This information is summarized for different laser rangefinder maximum distances in Table 1. As expected, the shorter is this distance, the greater is the probability to get the robot lost.

Tables 2 and 3 present the computed mean errors of all the trials for each laser configuration with respect the trajectory ground-truth for both, Plain MCL and MCL with Particle Injection. It can be seen that the errors stay close to 1 m. in position and 0.05 rad. in orientation for all the laser configurations for the proposed algorithm, while the errors in position are very large in the case of the plain MCL when laser is limited to 5 and 10 meters.

Our proposal has also been tested to guess the initial position of the robot, as can be seen in Fig. 9. When the initial position is unknown the localization of the robot is randomly set by MCL. In this case, the BoW plus robust matching algorithms are the key in order to properly re-localize the robot at the correct position when a first match is encountered, which is equivalent to restarting the filter with the estimated current position. A new sample set is

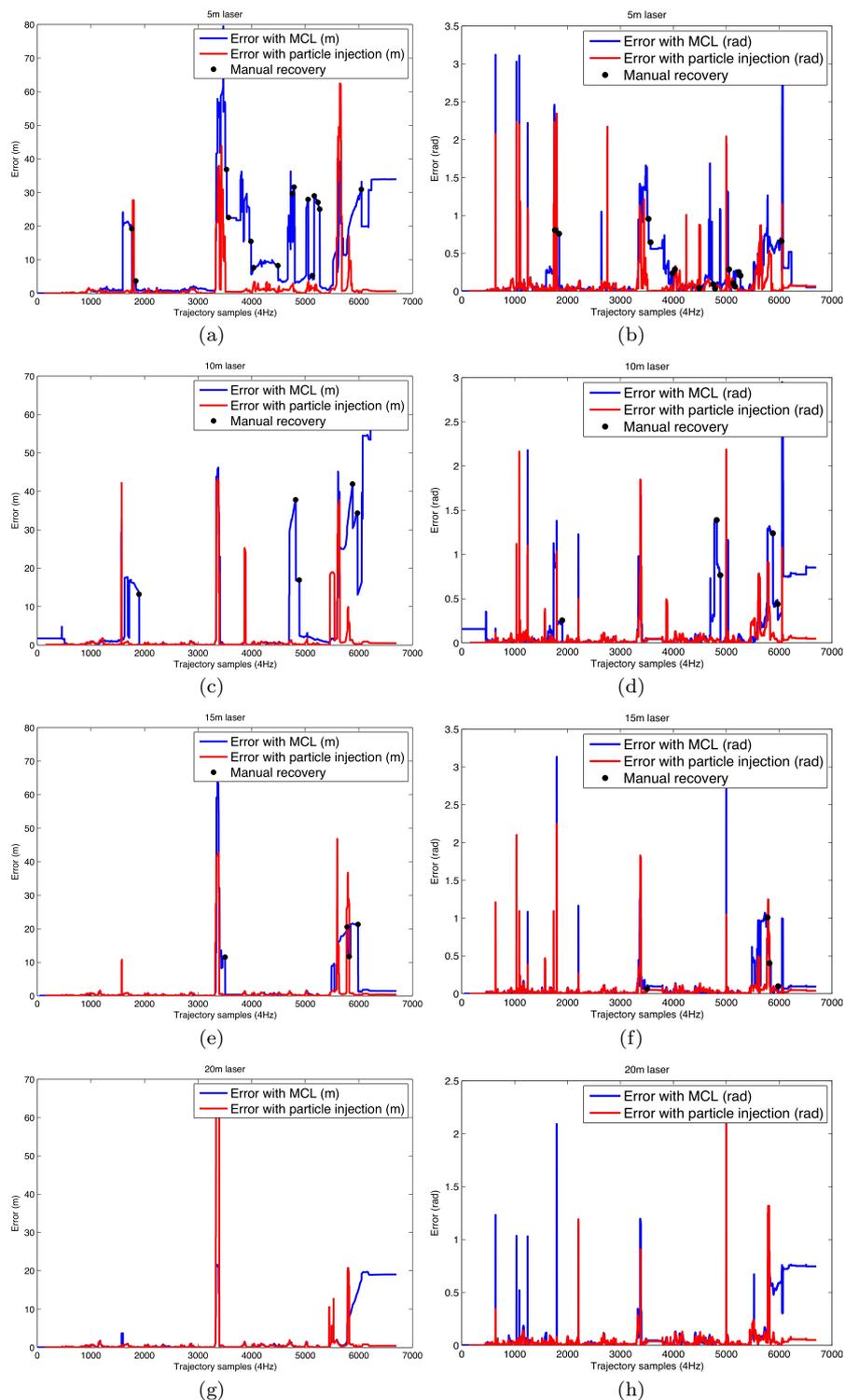


Fig. 8 Mean position error (left) and mean absolute orientation error (right) comparison between MCL (blue) and Particle Injection Algorithm (red) of 6 simulations done for each laser configuration. MCL was supervised and manually recovered when lost. All manual recoveries are represented in these graphics (a point is shown if at least 1 of the 6 simulations got lost at that point). It can be seen the impact on the mean errors. Mean manual recoveries are presented in Table 1 for each laser configuration.

Table 3 Mean and Std. deviation for orientation error for both MCL and MCL + Particle Injection against ground truth pose

Max Laser Range	Orientation error (rad.)	
	MCL	MCL + Particle Injection
5m	0.20 ± 0.33	0.08 ± 0.19
10m	0.20 ± 0.35	0.05 ± 0.14
15m	0.08 ± 0.19	0.04 ± 0.13
20m	0.12 ± 0.24	0.037 ± 0.10

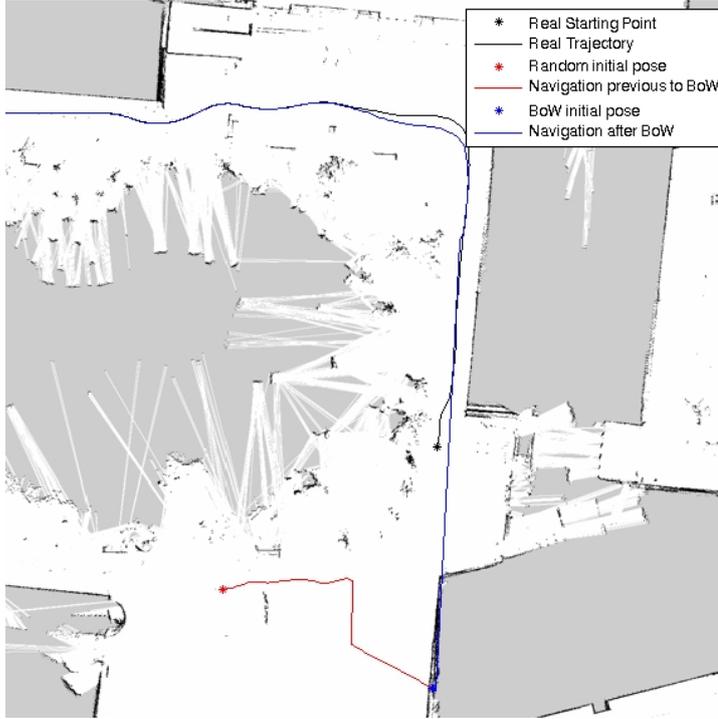


Fig. 9 Robot localization starting at a unknown position (kidnaping problem). Blue solid line: Estimated robot trajectory using MCL+BoW. Black solid line: True position of the robot. Red solid line: MCL localization when robot is started with random position.

generated with the first good match, taking the position of the matched image as the correct localization, allowing the module to restart the filter at this point, converging to the real trajectory of the robot.

5.2 Evaluation at Royal Alcázar

The proposed method has been also tested in the Real Alcázar of Seville in a 2 km-long tour, and with a duration of half an hour approximately (the map and the trajectory followed by the robot is shown in Fig. 10). This environment presents significant daylight variations due to shadows of buildings, and it is

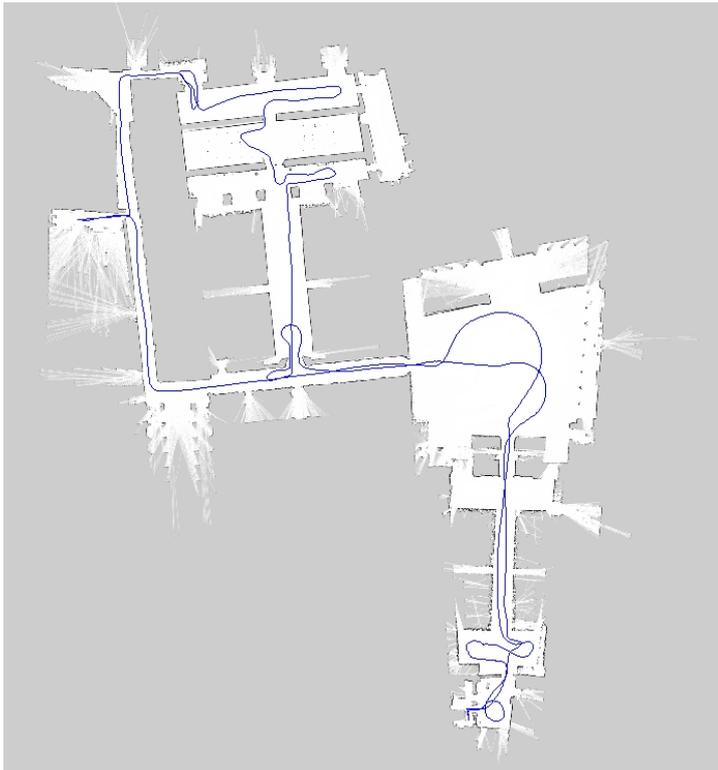


Fig. 10 Map and ground truth trajectory of FROG robot at Royal Alcázar of Seville.



Fig. 11 Crowded and dynamic environment may disturb classic scan matching localization. The figure shows the robot completely surrounded by students. This situation is common in very tourist areas like Royal Alcázar

much more crowded than Lisbon Zoo, making this a highly variant scenario (see Fig. 11). A map was generated as detailed in Section 3. This map has been augmented with appearance information obtained from 5 complete image datasets taken at different hours of the day (see Fig. 12). The resulting working database contains 971 elements, defined by the BoW representation of images of interest and the poses of the robot within the map where those images were collected.



Fig. 12 Different scenes at Royal Alcázar, showing daylight variation of images used to create database.

With this new database our algorithm is used to guess the initial pose of the robot. As can be seen in Fig. 13, MCL sets a random initial pose (black asterisk) and after 20cm of navigation our proposal detects a good first match (see Fig. 14), setting a new initial pose and reinitializing the particle filter hypothesis (red asterisk) very close to ground truth initial pose (blue asterisk), with an error in position less than 1.5m and less than 0.131rad in orientation. After the initial pose is set, our algorithm will never reinitialize the particle filter; instead of that, the Particle Injection method will substitute less significant particles within the current hypotheses with new ones at the position of the matched database image (see Algorithm 2). As Fig. 15 shows, new particles will not affect the localization hypothesis if previous particles have a better localization, in this case our algorithm discards new particles in resampling process as commented before. Normal execution of the algorithm in Royal Alcázar presents positive matches each 3m of navigation, while worst case scenario (areas without enough features) presented positive matches each 10m of trajectory.

In Fig. 15 is likewise shown the behavior of the Particle Injection method when kidnapping the robot by manually localizing the robot in a wrong pose and making the filter converge to this wrong location. The new particles injected have a higher likelihood for laser measurements (see lines 15 to 20 Algorithm 1) and provoke the correct relocalization of the system.

6 Conclusions and Future Work

The paper presented an algorithm to integrate visual place recognition and Monte Carlo Localization in order to provide a more robust localization of a mobile ground robot. The method is based on injecting new pose hypotheses



Fig. 13 Black asterisk: Random initialization of MCL at Royal Alcázar of Seville. Blue asterisk: Ground truth initial pose. Red asterisk: Particle Injection first match: reinitialization of MCL hypothesis to this point. The subsequent estimated trajectory (red) is very close to the ground truth trajectory (blue).

from the appearance information, which are then checked by the measured laser. The method, on one hand, may reduce slightly the accuracy of the filter, as the variance of particles is increased. However, this is compensated by an increase of the robustness of the system, alternative hypotheses are continuously evaluated.

The experimental results with datasets in the Lisbon Zoo and Royal Alcázar of Seville, mixed indoor-outdoor challenging scenarios (see Figs. 7 and 10), show that the method behaves correctly and dramatically reduce the mean number of times the robot get lost, with the corresponding impact in position accuracy and reliability.

The effect of different degrees of occlusions has been emulated by artificially restricting the maximum range of the laser to different values. While this effect is not exactly the same as sporadic occlusions by people, it allows a comparisons of the approaches, keeping other aspect untouched. Furthermore, the minimum range employed means that the laser is not able to see any of the related features in the robot map in some places for both the scenarios, emulating a complete occlusion.

Future work related with this algorithm may include the analysis of the navigation area and the use of different models for the distribution of the newly injected particles in different areas of the map. In this approach, Gaussian models with the same parameters has been used as distributions; these parameters summarize the global accuracy of the map building process. But in corridors, for instance, it should be more efficient use of distributions that concentrate their uncertainty along them, making the distribution more elongated in the direction of corridor and shorter in the cross direction. The same analysis can be done for other areas, taking into account how people walk and distribute within this crowded environment, in areas like patios, large rooms and other points of interest and also considering information about typical planned tours for visitors.

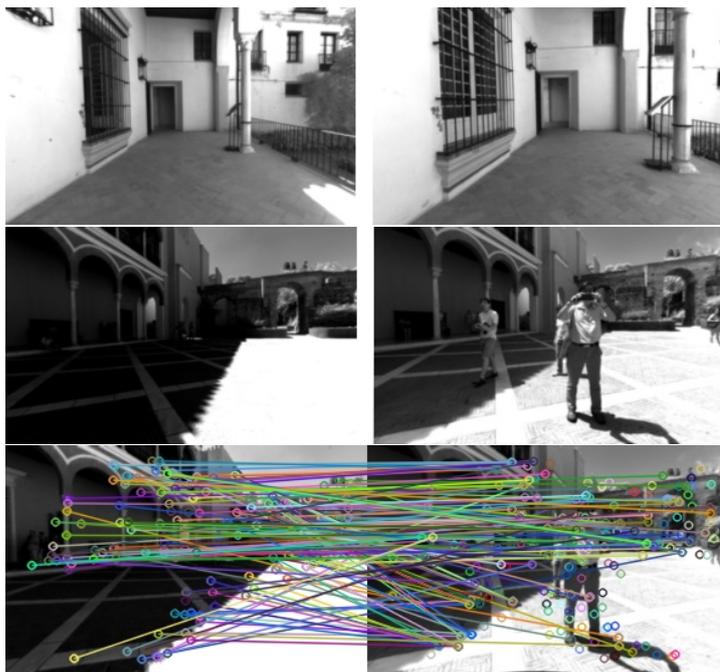


Fig. 14 Top: First match at Royal Alcázar of Seville (current image on the left, matched image on the right). At this moment, the MCL is reinitialized with the position where the matched image was recorded. Middle and bottom: matching with people and daylight variation in scenes. Robust Matching based on Fundamental matrix test is applied over OpenFabMap2 to discard wrong matches.

References

1. Alcantarilla, P.F., Stasse, O., Druon, S., Bergasa, L.M., Dellaert, F.: How to localize humanoids with a single camera? *Auton. Robots* **34**(1-2), 47–71 (2013). DOI 10.1007/s10514-012-9312-1. URL <http://dx.doi.org/10.1007/s10514-012-9312-1>
2. Angeli, A., Filliat, D., Doncieux, S., arcady Meyer, J.: A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM* pp. 1027–1037 (2008). URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.146.5171>
3. Carlone, L., Bona, B.: A comparative study on robust localization: Fault tolerance and robustness test on probabilistic filters for range-based positioning. In: *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–8 (2009)
4. Corke, P., Paul, R., Churchill, W., Newman, P.: Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation. In: *IROS*, pp. 2085–2092. IEEE (2013)
5. Dayoub, F., Duckett, T.: An adaptive appearance-based map for long-term topological localization of mobile robots. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 3364–3369 (2008). DOI 10.1109/IROS.2008.4650701
6. Doucet, A., Freitas, N.d., Murphy, K.P., Russell, S.J.: Rao-blackwellised particle filtering for dynamic bayesian networks. In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI '00*, pp. 176–183. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (June, 2000). URL <http://dl.acm.org/citation.cfm?id=647234.720075>

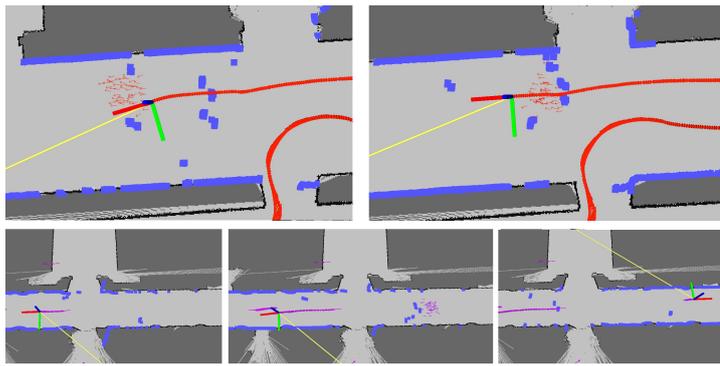


Fig. 15 Top: Particles injected while navigating. It is shown that particles injected will not affect localization hypothesis if previous particles are better localized. Bottom: Particles injected while kidnapping. The robot is localized in the same corridor but in wrong direction, making the filter converge to this wrong hypothesis. As can be seen, new particles injected but bad localized are discarded while new ones better localized make filter favour them and penalize old particles recovering localization.

7. Galvez-Lopez, D., Tardos, J.D.: Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics* **28**(5), 1188–1197 (2012). DOI 10.1109/TRO.2012.2197158
8. Glover, A.J., Maddern, W.P., Warren, M., Reid, S., Milford, M., Wyeth, G.: Openfabmap: An open source toolbox for appearance-based loop closure detection. In: *ICRA*, pp. 4730–4735. IEEE (2012)
9. Hentschel, M., Wagner, B.: An adaptive memory model for long-term navigation of autonomous mobile robots. In: *Journal of Robotics* (2011)
10. Himstedt, M., Keil, S., Hellbach, S., Böhme, H.J.: A Robust Graph-based Framework for Building Precise Maps from Laser Range Scans. In: *ICRA, Workshop on Robust and Multimodal Inference in Factor Graphs* (2013)
11. Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g2o: A general framework for graph optimization. In: *Proc. International Conference on Robotics and Automation, ICRA*, pp. 3607–3613. IEEE (2011). URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5979949
12. Kümmerle, R., Ruhnke, M., Steder, B., Stachniss, C., Burgard, W.: A navigation system for robots operating in crowded urban environments. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3225–3232 (2013)
13. Pérez-Lara, J., Caballero, F., Merino, L.: Long-term ground robot localization architecture for mixed indoor-outdoor scenarios. In: *Proceedings of the International Symposium on Robotics, ISR*
14. Pérez-Lara, J., Caballero, F., Merino, L.: Integration of Monte Carlo Localization and Place Recognition for Reliable Long Term Robot Localization. In: *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC 2014)*. Espinho, Portugal (2014)
15. Ruhnke, M., Kümmerle, R., Grisetti, G., Burgard, W.: Highly accurate maximum likelihood laser mapping by jointly optimizing laser points and robot poses. In: *Proc. International Conference on Robotics and Automation, ICRA* (2011)
16. Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Hahnel, C.: Probabilistic algorithms and the interactive museum tour-guide robot minerva. *The International Journal of Robotics Research* **19**, 972–999 (2000)
17. Thrun, S., Burgard, W., Fox, D.: *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press (2005)
18. Thrun, S., Fox, D., Burgard, W., Dellaert, F.: Robust monte carlo localization for mobile robots. *Artificial Intelligence* **128**(1-2), 99–141 (2000)

-
19. Wallach, H.M.: Topic modeling: Beyond bag-of-words. In: Proceedings of the 23rd International Conference on Machine Learning, ICML '06, pp. 977–984. ACM, New York, NY, USA (2006). DOI 10.1145/1143844.1143967. URL <http://doi.acm.org/10.1145/1143844.1143967>