

Integration of Monte Carlo Localization and Place Recognition for Reliable Long-Term Robot Localization

Javier Pérez¹, Fernando Caballero² and Luis Merino¹

Abstract— This paper proposes extending Monte Carlo Localization methods with visual information in order to build a long term 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 with 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.

I. INTRODUCTION

The work of this paper is part of larger project, the project FROG³, 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 [1], 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 demonstrate a long-term operation of the robot in the Lisbon Zoo and the Royal Alcazar in Seville (see Figure 1).

Navigating in these crowded places (the Royal Alcazar 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, etc. 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 [2].

Scan matching approaches based on 2D lasers are the most extended localization algorithms, due to their high accuracy compared to other sensors like ultrasonic sensors, and with a low processing cost compared to vision sensors [3]. 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

This work is partially funded by the European Commission 7th Framework Programme under grant agreement no. 288235 (FROG) and the project PAIS-MultiRobot, funded by the Junta de Andalucía (TIC-7390)

¹ Javier Pérez and Luis Merino are with the Pablo de Olavide University, Seville (Spain) {jiperlar, lmercab}@upo.es

² Fernando Caballero is with University of Seville, Seville (Spain) fcaballero@us.es

³ <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 Alcazar of Seville and the Zoo of Lisbon. Young visitors surrounding the robot in the Royal Alcazar, interfering in sensor readings and interrupting robot's trajectory.

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 Alcazar, where people may approach and surround the robot driven by curiosity (see Figure 1) or while they are being guided.

Several approaches have been considered to enhance the robustness of localization systems. Thus, Hentschel and Wagner [4] 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 [6], DLoopDetector [7], and others [8] that use structures based on Bag-of-Words [9] have been presented to look for revisited places, what is helpful for recovering from localization errors. Corke et al. [10] 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 tests. Then, Section

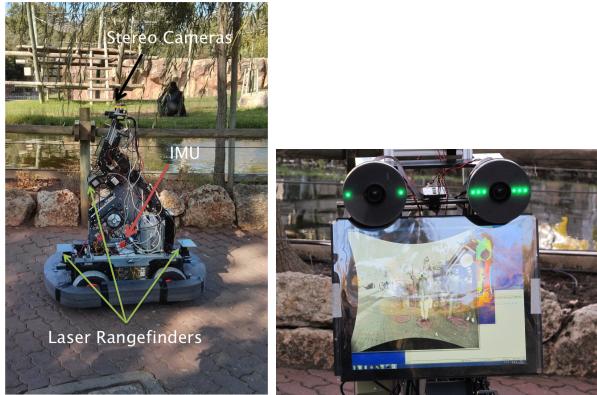


Fig. 2. Left: the FROG robot platform. It shows the main platform and the positions of the sensors. Right: the stereo pair located in the robot eyes and the touch screen. It can be seen the estimation of persons poses on the screen.

III presents how ground truth pose was obtained for map building and for testing our algorithm described in Section IV for long-term-localization. Paper ends with Section V detailing experiments done and results obtained and Section VI details conclusions and future work.

II. THE ROBOT PLATFORM

Figure 2 shows a picture of the robot considered as deployed in the Lisbon Zoo 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 MTi-G IMU from XSense
- Three laser rangefinders are considered. Two deployed horizontally forward and backwards, employed for localization and obstacle avoidance. The third laser is used for 3D perception of slopes
- An stereo pair, employed for person detection, robot pose estimation and 3D perception.
- An additional camera is used for low-range affective computing of the interacting persons

III. MAP BUILDING

To test the proposed algorithm it is necessary to get a map of the navigation area with high accuracy. The real experiments shown in this paper were conducted at the Lisbon Zoo. Being a GPS-denied place, this scenario requires a SLAM solution for building an accurate map.

However, the application considered allows for an offline SLAM solution: the robot can be deployed in the scenario

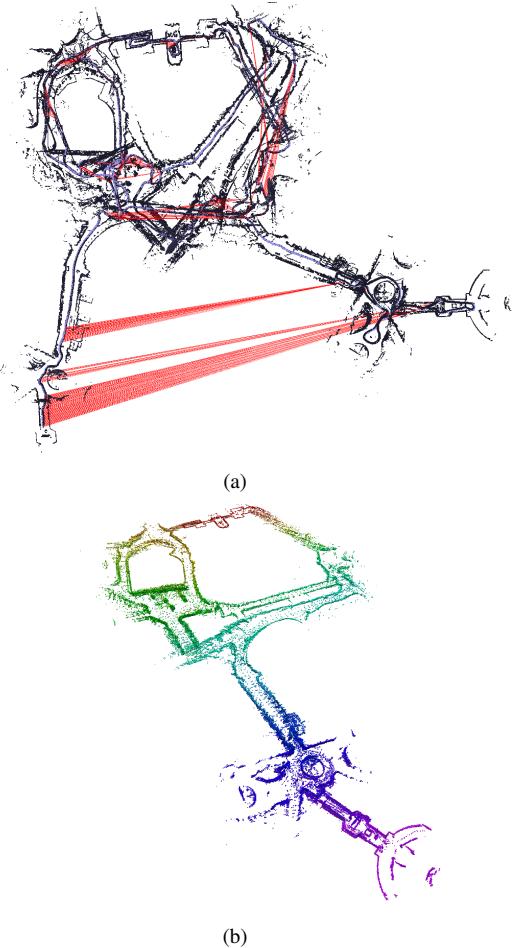


Fig. 3. (a) Loop closures obtained at the Lisbon Zoo (red lines, between revisited places). It can be also appreciated the typical drift associated to odometry. (b) Point cloud obtained from the Lisbon Zoo. The height of the points is color-coded (reddish colors indicating higher ground).

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, consists of 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 backend. Our front-end considers odometry and loop closures provided the algorithm OpenFabMap2 [6] over the images to provide constraints on the state variables, in this case the robot poses.

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 data) in the global frame will lead to maps with slight errors, such as fuzzy walls or double walls, as the information from those sensors was not directly considered in the minimization process. Therefore, a final procedure is used to optimize the resulting map. The following steps are carried out:

- 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.
- 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.

Figure 4 shows an example of this refinement. The final resulting map after 2D laser scan integration is shown in Figure 5. A comparison of laser scan integration and CAD map is shown in Figure 6. 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 Figure 5, non-permanent obstacles like pedestrians are eliminated due to data integration when building the map (only consistent obstacles are included).

IV. LONG-TERM LOCALIZATION ALGORITHM

A. 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 [12]. 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

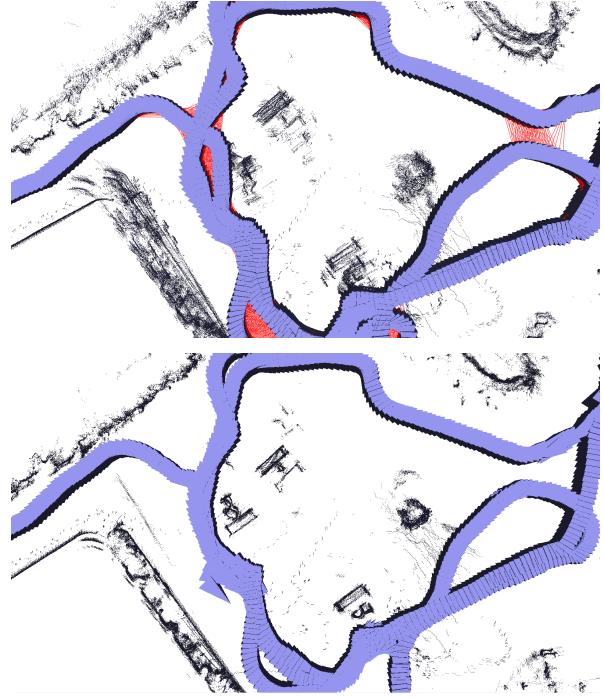


Fig. 4. Top: map obtained by projecting the information with the optimized robot pose. Red lines refer to loop closure detection. Bottom: Map refinement by considering the laser information into the optimization process. It can be seen how the grass on the top left is correctly aligned. Also, some trees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

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 [13], 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 (see Algorithm 1, lines 6 to 9 for odometry prediction, and 17 to 18 for updates). 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.

B. Appearance-based particle injection

The resampling process and occlusions of laser rangefinders may introduce errors in localization in certain executions [14], causing the particle filter to diverge or to converge to wrong locations. Intelligent re-sampling techniques can be used to limit these effects [15], but they cannot be



Fig. 5. Resulting occupancy grid map of Lisbon Zoo after 2D laser scan integration.

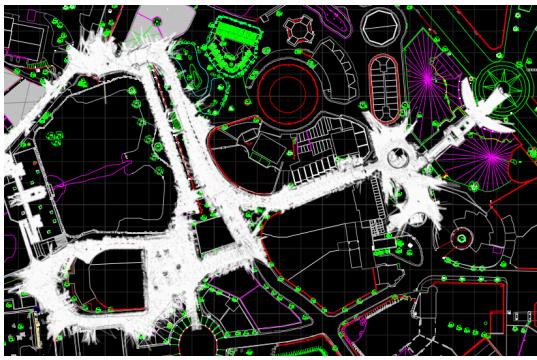


Fig. 6. 2D occupancy map of the Lisbon Zoo overlaid on a CAD drawing of the Zoo.

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 [6]. 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 idea. During the mapping stage detailed in Section III, 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 images I_t and image I_k of the database is detected (Figure 7), 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

Algorithm 1 Base Localization Algorithm

```

1:  $\langle x_t^{[i]}, y_t^{[i]}, \theta_t^{[i]}, \bar{z}_t^{[i]}, \sigma_{z,t}^{[i]}, \gamma, \varphi, \omega^{[i]} \rangle_i^L$  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:    $\langle x_{t+1}^{[i]}, y_{t+1}^{[i]}, \theta_{t+1}^{[i]} \rangle \leftarrow \text{sample\_kinematic\_model}$ 
7:    $(x_t^{[i]}, y_t^{[i]}, \theta_t^{[i]}, \mathbf{u}_t, \Delta t)$ 
8:    $\mathbf{v}_g^{[i]} = \mathbf{R}(\gamma, \varphi, \theta^{[i]}) [v \ 0 \ 0]^T$ 
9:    $\hat{z}_{t+1}^{[i]} = \bar{z}_t^{[i]} + \Delta t \mathbf{v}_g^{[i]}$ 
10:   $\hat{\sigma}_{z,t+1}^{[i]2} = \bar{\sigma}_{z,t}^{[i]2} + \sigma^2$ 
11:   $\bar{z}_{t+1}^{[i]} = \bar{z}_t^{[i]} - \frac{\hat{\sigma}_{z,t+1}^{[i]2}}{\hat{\sigma}_{z,t+1}^{[i]2} + \sigma_{z,M}^2} (z_t^{[i]} - h_M(x_t^{[i]}, y_t^{[i]}))$ 
12:   $\hat{\sigma}_{z,t+1}^{[i]2} = \frac{\hat{\sigma}_{z,t+1}^{[i]2} \sigma_{z,M}^2}{\hat{\sigma}_{z,t+1}^{[i]2} + \sigma_{z,M}^2}$ 
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

```

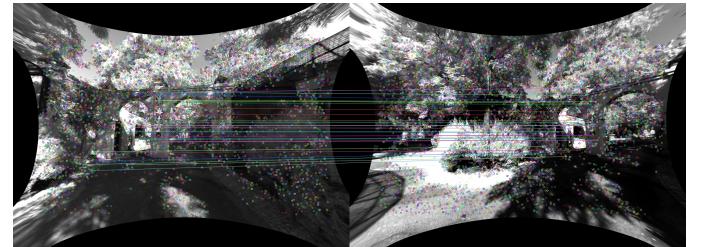


Fig. 7. Matching between images.

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 tests done), 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 tests).

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 of the laser

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 dis-
         tributed with center pose  $\langle x_k, y_k, \theta_k \rangle$  and weight
          $W_{injected}$ 
8:     end if
9:   end if
10: end if

```

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 (in Figure 2 can be seen that recovery time is in order of 17 secs in worst case).

V. EXPERIMENTAL RESULTS

This section details a set of experiments conceived to validate the proposed method. The robot sensors (lasers, cameras, odometry, ...) were recorded for one hour approximately while navigating 1.5 km. at the Lisbon Zoo, a challenging scenario with slopes, vegetation and non-structured information. This dataset is different than the one used to build the map and the position-tagged images for place recognition. The experiments will evaluate how the localization accuracy is improved when visual place recognition is integrated into the MCL algorithm.

In order to measure the localization accuracy of the algorithms, a ground truth robot pose is computed by using two 2D laser scanners covering 360 degrees surrounding the robot and executing MCL using 5.000 particles. The proposed algorithm (Particle Injection) and the base localization algorithm (called plain MCL) are tested against this ground truth data using the front laser and the cameras onboard. Different low cost lasers and the effect of crowded scenarios are simulated by decreasing the maximum range of the lasers. Tests are done by running 6 times both algorithms for one single frontal laser of 5m, 10m, 15m and 20m as maximum range.

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

TABLE I
MEAN MANUALLY RECOVERIES (TIMES PER TEST)

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

TABLE II
MEAN AND STD. DEVIATION FOR POSITION AND ORIENTATION ERROR
FOR MCL

Max Laser Range	MCL	
	position error (m.)	orientation error (rad.)
5m	10.73 ± 13.64	0.20 ± 0.33
10m	10.39 ± 19.74	0.20 ± 0.35
15m	2.27 ± 6.87	0.08 ± 0.19
20m	2.83 ± 6.18	0.12 ± 0.24

necessary in every configuration test, while our particle injection algorithm did not need to be manually recovered.

It can be seen in Figure 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 I. As expected, the shorter is this distance, the greater is the probability to get the robot lost.

Tables II and III presents the computed mean errors of all the trials for each laser configuration with respect the trajectory ground-truth for both, Plain MCL (Table II) and with particle injection (Table III). It can be seen that the errors stay close to 1 m. in position and 0.01 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.

VI. 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 the robot in crowded and non-planar scenarios. The method allows injecting particles in the close loop candidates, checking the position hypotheses with the measured laser.

The experimental results with datasets 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.

Future work related with this algorithm may include the analysis of navigation area and the realization of different models of distribution probability for particles in different areas of the map. In this approach Gaussian model has been used as distribution, but in corridors should be more efficient use of distributions that grow along them, making distribution more elongated in the direction of corridor and shorter in the cross direction. Same analysis can be done

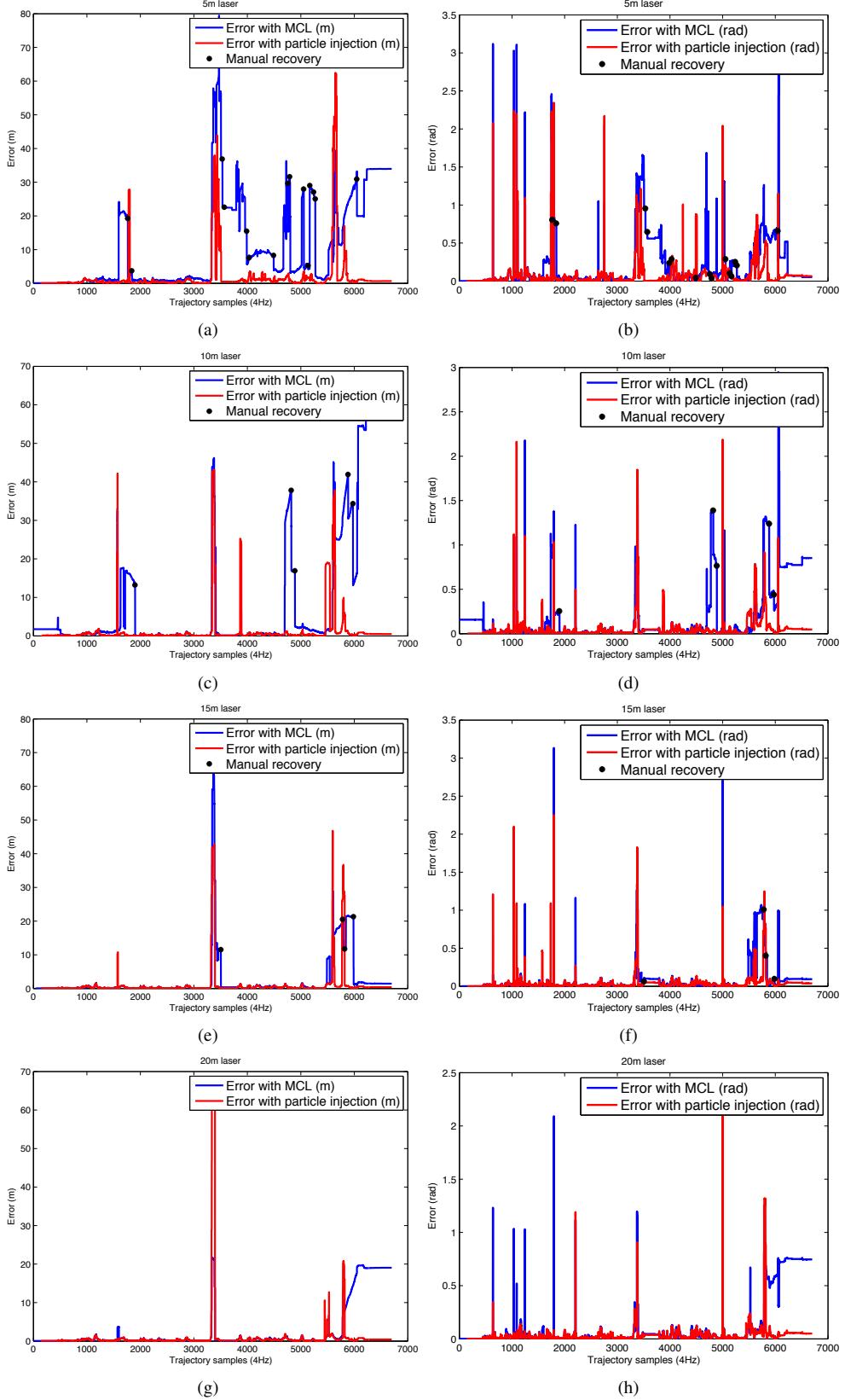


Fig. 8. Mean position error (left) and mean absolute orientation error (right) comparison between MCL (blue) and Particle Injection Algorithm (red) of 6 simulations done for each laser configuration. MCL was supervised and manually recovered when lost. All manual recoveries are represented in these graphics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

TABLE III
MEAN AND STD. DEVIATION FOR POSITION AND ORIENTATION ERROR
FOR AMCL + PARTICLE INJECTION

Max Laser Range	AMCL + Particle Injection	
	position error (m.)	orientation error (rad.)
5m	1.95 ± 6.59	0.08 ± 0.19
10m	1.27 ± 5.14	0.05 ± 0.14
15m	1.04 ± 4.85	0.04 ± 0.13
20m	1.10 ± 6.55	0.037 ± 0.10

for other areas, taking into account how people walk and distribute in this crowded environment, areas like patios, large rooms and others of interest and also with information about typical planned tours for visitors.

REFERENCES

- [1] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, and C. Hahnel, “Probabilistic algorithms and the interactive museum tour-guide robot minerva,” *The International Journal of Robotics Research*, vol. 19, pp. 972–999, October 2000.
- [2] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, “A navigation system for robots operating in crowded urban environments,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.
- [3] P. F. Alcantarilla, O. Stasse, S. Druon, L. M. Bergasa, and F. Dellaert, “How to localize humanoids with a single camera?” *Journal of Autonomous Robots*, 2012.
- [4] M. Hentschel and B. Wagner, “An adaptive memory model for long-term navigation of autonomous mobile robots,” in *Journal of Robotics*, 2011.
- [5] F. Dayoub and T. Duckett, “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*, Sept 2008, pp. 3364–3369.
- [6] A. J. Glover, W. P. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth, “Openfabmap: An open source toolbox for appearance-based loop closure detection,” in *ICRA*. IEEE, 2012, pp. 4730–4735.
- [7] D. Galvez-Lopez and J. D. Tardos, “Bags of binary words for fast place recognition in image sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.
- [8] A. Angeli, D. Filliat, S. Doncieux, and J. arcady Meyer, “A fast and incremental method for loop-closure detection using bags of visual words,” *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, 2008.
- [9] H. M. Wallach, “Topic modeling: Beyond bag-of-words,” in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML ’06. New York, NY, USA: ACM, 2006, pp. 977–984.
- [10] P. Corke, R. Paul, W. Churchill, and P. Newman, “Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation.” in *IROS*. IEEE, 2013, pp. 2085–2092.
- [11] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o: A general framework for graph optimization,” in *Proc. International Conference on Robotics and Automation, ICRA*. IEEE, 2011, pp. 3607–3613.
- [12] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust monte carlo localization for mobile robots,” *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.
- [13] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, “Rao-blackwellised particle filtering for dynamic bayesian networks,” in *Uncertainty in Artificial Intelligence (UAI)*. San Francisco, CA: Morgan Kaufmann, 2000, pp. 176–183.
- [14] L. Carlone and B. Bona, “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*, June 2009, pp. 1–8.
- [15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.