2022 I-RIM Conference
October 7-9, Rome, Italy
ISBN: 9788894580532
DOI:10.5281/zenodo.7531326

Efficient 2D LIDAR-Based Map Updating For Long-Term Operations in Dynamic Environments

Elisa Stefanini[1,2], Enrico Ciancolini[3], Alessandro Settimi[3] and Lucia Pallottino[1]

***Abstract*—Long-time operations of autonomous vehicles and mobile robots in logistics and service applications are still a challenge. To avoid a continuous re-mapping, the map can be updated to obtain a consistent representation of the current environment. In this paper, we propose a novel LIDAR-based occupancy grid map updating algorithm for dynamic environments. The proposed approach allows robust long-term operations as it can detect changes in the working area even in presence of moving elements. Results highlighting map quality and localisation performance, both in simulation and experiments, are reported.**

I. Introduction

Today, companies of all sizes use Autonomous Mobile Robots (AMR) to improve their productivity, and robotics is becoming increasingly important for future developments in the sector. An AMR employs Simultaneous Localisation and Mapping (SLAM) techniques where the robot maps the environment, navigates, and locates itself by simply "looking" at this environment, without the need of installing further hardware in the work place. Conventional SLAM is used to create an initial image of an unknown environment, and it is not intended to be used as a system for repeatedly updating the same area of the map. This approach could lead to task failure in complex production environments where changes in the map can occur. Industrial environments are usually characterized by a slow rate of change over time since semi-static objects could change their position when the same task is performed at another time, be it a day or two months later. During AMRs operations, small changes in the environment are well handled by localisation and local obstacle avoidance algorithms; however, as the robot's static map diverges from the current working area, becoming obsolete, navigation performance degrades [1].

In this work, as also described with more details in [2], we propose a system based on 2D LIDAR measurements capable of detecting changes in the environment over time and updating an existing map with a memory-limited algorithm taking into account localisation and measurement errors while neglecting highly dynamic obstacles as humans. Our system is fully developed through the Robot Operating System, using a 2D occupancy grid map representation. We are interested in taking as input such map type built at any moment with any available SLAM algorithm and producing a geometrically and temporally consistent representation of the environment suitable for any continued localisation algorithm based on occupancy grid maps.

[1]Centro di Ricerca "E. Piaggio" e DII, Università di Pisa, Largo L. Lazzarino 1, Pisa, Italy.

[2]SoftBots, IIT, via Morego, 30, Genova, Italy

[3]Proxima Robotics s.r.l., Via Olbia 20, Cascina, Pisa, Italy

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement Number 101017274 (DARKO), and in part by the Italian Ministry of Education and Research (MIUR) through the CrossLab Project (Departments of Excellence).

II. Map Updated Method

Referring to the system overview in Fig.1, given an initial occupancy grid map M and the robot pose $X(k)$, the basic idea is to update the state of the cells according to the relevant changes in the environment detected by the laser measurements $Z(k)$. The system is built to detect both the removal or addition of static objects while neglecting the presence of dynamic obstacles, that can be sources of disturbances. Measurements are processed in two steps. First, the *Beams Classifier* analyses the sensor readings $z_i(k)$ and classifies them as "detected change measurement" or "non-detected change measurement" according to their discrepancy w.r.t. the initial map M. Then the *Changed Cells Evaluator* and the *Unchanged Cells Evaluator* evaluate cells associated to measurements $z_i(k)$ w.r.t. those in the initial map M to confirm the type of detection. To avoid false changes or undetected ones, we don't change the state of the cells based only on one measurement. Indeed, for each cell c_j, a rolling buffer, B_{c_j}, of a fixed dimension, N_b, is created and filled with the outcomes of the evaluator blocks at different time instant. The *Changed Cells Evaluator* takes as input only measurements $z_i(k)$ for which the *Beams Classifier* has "detected" a change and fills the buffer B_{c_j} for each associated cell, c_j, with a "changed" flag only if the change is confirmed. Instead, the *Unchanged Cells Evaluator* analyses only measurements $z_i(k)$ for which the *Beams Classifier* provides a "non-detected" outcome and fills the buffer B_{c_j} for each associated cell, c_j, with an "unchanged" flag only if the evaluation is confirmed. Finally, the state of evaluated cells c_j is changed from free to occupied (or vice versa) only if a sufficiently high number of "changed" flags can be found in the associated buffer, B_{c_j}.

III. Experiments

In this section, we present the results of our approach both in simulation and on real-world data. To provide a quantitative performance evaluation of the system, we compared our updated maps with ground-truth ones using the Cross-correlation (CC), the Map Score (MS), and the Occupied Picture-Distance-Function (OPDF) metrics [3], [4]. Moreover, we analysed the localisation errors with and without our updated maps using the Evo Python Package[1] in the simulation experiments.

A. Simulation experiments

We built four different versions of an industrial warehouse of 290 m^2 by increasing the environment changes to simulate how the placement of goods within a warehouse can change

[1]https://github.com/MichaelGrupp/evo

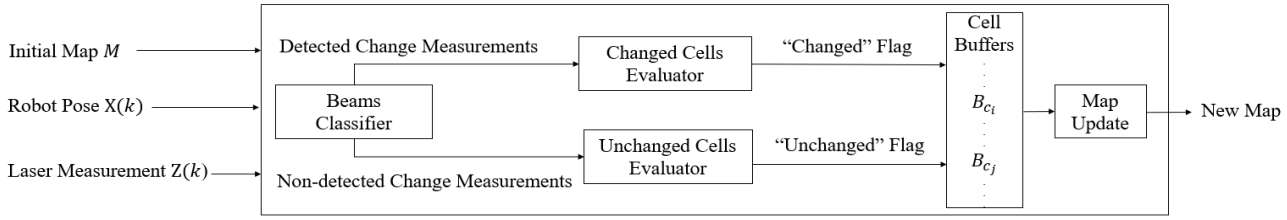

Fig. 1: System Overview: Measurements are classified as "detected change" or "non-detected change" w.r.t. the initial map by the *Beam Classifier*. Based on this classification, a rolling buffer B_{c_j} of each cell c_j is filled by the *Changed Cells Evaluator* and the *Unchanged Cells Evaluator* respectively through "changed" and "unchanged" flags. Finally, the state of the cells is updated if the number of "changed" flags in the buffer is higher than a given threshold.

over time. In the first environment W_1, a Summit-XL-Steel platform, was tele-operated to build an adequate initial map M_1. In the other environments W_i, the robot autonomously performed the same predefined trajectory. In each scenario W_i, the robot used the map M_{i-1} as the initial map to localise itself, and generated the updated map M_i with the proposed updating method. To evaluate the quality of the method a ground-truth maps G_i for each run$_i$ has been obtained with Slam Toolbox. Adaptive Monte Carlo Localisation (AMCL) from ROS have been used for robot localisation.

	W_2		W_3	
	M_1/G_2	M_2/G_2	M_1/G_3	M_3/G_3
CC (%)	45.05	69.26	31.78	60.61
MS (%)	48.35	70.66	36.66	61.70
OPDF (%)	66.08	95.64	53.25	91.32

TABLE I: Quantitative maps evaluation.

1) Updating Performance: The Table I shows the quantitative results obtained by comparing both the initial map M_1 and our updated maps $M_i, i \in \{2, 3\}$ w.r.t. the ground-truth ones G_i where a 100% score is a full correspondence of the two maps. A map comparison between M_{i-1} and G_i is performed to quantify differences between current and previous environment (first column for each W_i). According to each metric, the updated map M_i is always better than the initial map M_{i-1} of the run when compared with the ground-truth.

2) Localisation Performance: To evaluate improvements in localisation performance thanks to the use of our maps, we compared the AMCL pose estimate based on both the initial map M_1 and the last available updated map with the reference ground truth obtained from the simulator. Quantitative results of the localisation performance obtained in W_3 are reported in Tab. II where is shown that the localisation is drastically improved thanks to the use of the updated map M_2.

		Max	Mean	Median	Min	RMSE	SSS	Std
W_3	M_1	1.05	0.31	0.08	0.03	0.50	113.90	0.35
	M_2	0.12	0.05	0.05	0.03	0.05	1.57	0.01

TABLE II: Localisation Performance comparison in W_3 using both the initial map M_1 and the last updated map M_2.

3) Hardware Resource Consumption: The computation of the CPU percentage and memory MB usage obtained through the ROS package "cpu monitor" in the map updating and localisation phase reported a mean of 15% for CPU and 55-53 Mb for the memory. Thus, the proposed memory-limited solution is suitable for life-long operation scenarios.

B. Real Experiments

To demonstrate the real-world applicability of the proposed method, we repeated the same procedure of the simulated experiments in the CrossLab lab of the Pisa Information Engineering Department, with a Summit-XL-Steel platform. Performance on map update and hardware resource consumption have been quantified as in the simulations. For the localisation performance, since no ground-truth external tracking system was available, it was not possible to provide valid and scientifically acceptable performance results.

1) Updating Performance: The quantitative results are shown in Table III. Considerations in Sub-section III-A1 regarding the metrics results are still valid here.

	W_2		W_3	
	M_1/G_2	M_2/G_2	M_1/G_3	M_3/G_3
CC (%)	69.69	75.98	63.77	68.80
MS (%)	54.63	64.53	51.44	57.36
OPDF (%)	84.61	95.05	78.92	90.77

TABLE III: Quantitative maps evaluation.

2) Hardware Resource Consumption: The CPU and memory usage are lower respectively from 15% to 5% and from 55-58Mb to 51-54Mb w.r.t. the simulated environments because of the decrease in no. of updated cells since the real environment is smaller than the simulated one.

IV. Conclusions

We have shown how our method is able to update the map robustly, reflecting the environment configuration w.r.t. localisation and measurement errors, neglecting humans, and limiting memory storage both in simulation and with real-world experiments. As future work, we plan to validate the localization performance in a real environment with an external tracking system.

References

[1] M. Dymczyk, *et al.*, "Map summarization for tractable lifelong mapping," in *RSS Workshop*, 2016.

[2] E. Stefanini, *et al.*, "Efficient 2d lidar-based map updating for long-term operations in dynamic environments," in *2022 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2022. [Online]. Available: https://www.dropbox.com/s/2o8mqbkza5tqwya/IROS_Efficient__2D_ _LIDAR__Based_Map_Updating__For_Long_Term_Operations_in_ Dynamic__Environments.pdf?dl=0

[3] O. Sullivan, "An empirical evaluation of map building methodologies in mobile robotics using the feature prediction sonar noise filter and metric grid map benchmarking suite," Master's thesis, University of Limerick, 2003.

[4] K. Baizid, *et al.*, "Vector maps: A lightweight and accurate map format for multi-robot systems," in *Int. Conf. Intell. Robots Syst.* Springer, 2016, pp. 418–429.