Local SLAM in Dynamic Environments using Grouped Point-Cloud Tracking. A Framework*

George Todoran[1], Markus Bader[2] and Markus Vincze[1]

***Abstract*— In the domain of mobile robotics, local maps of environments represent a knowledge base for decisions to allow reactive control, preventing collisions while following a global trajectory. Such maps are normally discrete, updated with relatively high frequency and no dynamic information. The proposed framework uses a sparse description of clustered scan points from a laser range scanner. Those features and the system odometry are used to predict the agent ego motion as well as the features motion, similar to a Simultaneous Localization and Mapping (SLAM) algorithm but with low-constraint features. The presented local Simultaneous Localization and Mapping (LSLAM) approach creates a decision base, holding a dynamic description which relaxes the requirement of high update rates. Experimental results demonstrate environment classification and tracking as well as self-pose correction in dynamic and static environments.**

I. INTRODUCTION

Local maps typically consist of close-vicinity representations of the environment. As they represent the closest layer of perception in relation with agent dynamic tasks (path-following, grasping etc.), they are required to be accurate, online and descriptor-rich. The presented framework proposes an alternative to local map-creation and environment description for mobile agents in an online, fast-computing manner. Thus, the environment is modelled and grouped as rigid dynamic objects, treating object discovery, time-out, merging, splitting and symmetries. Using the acquired information regarding the objects dynamic state, agent self-pose correction is performed, enchanting local map-building to local Simultaneous Localization and Mapping (LSLAM). In addition, the framework outputs the classified objects and their dynamic descriptors for further usage in self-localization, mapping, path-planning and other robotics tasks.

Grouping of data in higher level features–objects has been widely studied in computer vision and robotics communities and recently proposed in SLAM approaches [1]. However, this work aims to include high-level features in a more complex SLAM problem, where dynamic entities are present. Dynamic object tracking has been addressed by Montessano [3] in his PhD. thesis, analysing various filtering techniques. MacLachlan [2] presents a segmentation approach for occluded areas based on agent movement, here generalized for concave structures and used within the segmentation module.

*This work was not supported by any organization

[1]George Todoran and Markus Vincze are with the Automation and Control Institute, Vienna University of Technology, Gusshausstr. 27-29/E376, 1040 Vienna, Austria `[george.todoran, markus.vincze]@tuwien.ac.at`

[2]Markus Bader is with the Institute of Computer Aided Automation, Vienna University of Technology, Treitlstr. 1-3/4. Floor/E183-1, 1040 Vienna, Austria `markus.bader@tuwien.ac.at`

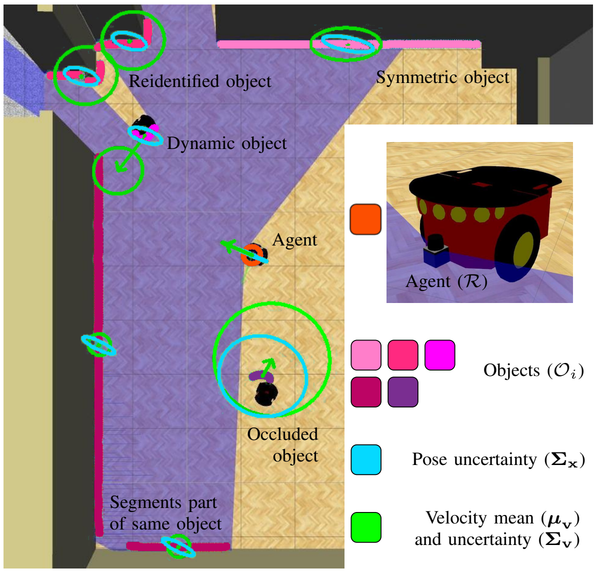

Fig. 1: Generated Local Map

II. APPROACH

As the agent $\mathcal{R}$ moves within an environment, the local environment is classified and mapped. Due to sensor characteristics and locality of the data association problem, the mapped point-clouds are represented in polar coordinates in the sensor frame, propagated in time with respect to the agent motion. However, their clusters are represented within the *LSLAM* estimator in a Cartesian space. Figure 2 presents the overview of the framework, including its modules and data-flow. In the following, the general characteristics and approaches towards each of the modules is presented.

a) Preprocessing: As the sensor input points $\hat{\mathbf{p}}_{raw}$ are modelled with noise in the range measurement, they are filtered in polar space using a continuous Gaussian-kernel. Given such sparse representation, reduced smoothing and eventually neglection of points close to a feature edge is achieved. Moreover, the module clusters the points in segments $\hat{\mathcal{S}}$, bounded by discontinuity regions.

b) Homographic Segmentation: As the entire world is assumed to be dynamic and of interest to the agent, the necessity of segmentation could be questioned. Overall, the purpose of segmentation in this framework is to reduce the number of points without correspondences in the matched point-clouds, increasing the robustness of *Covariant ICP*. Given the agent states $\boldsymbol{\mu}_{\mathcal{R}}$ and $\overline{\boldsymbol{\mu}}_{\mathcal{R}}$ when segments $\bar{\mathcal{S}}$ and $\hat{\mathcal{S}}$ are acquired, points that would not satisfy the ray assumption are segmented using occluded points removal, constant angular re-sampling and outlier removal.

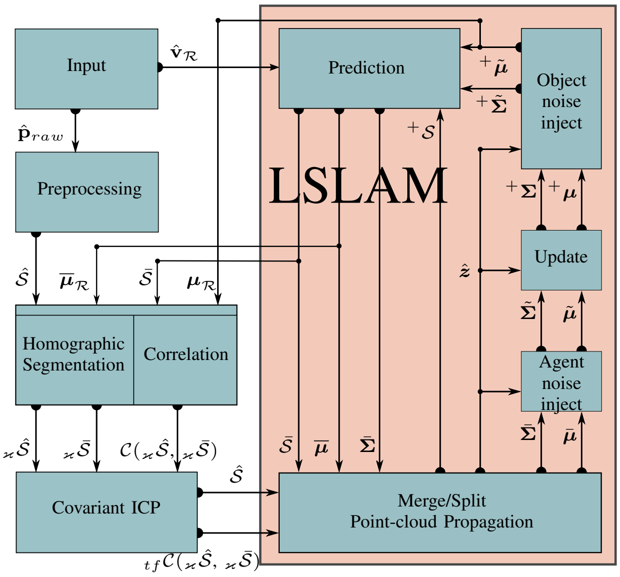

Fig. 2: Framework overview

c) Correlation: Using the same inputs, the module creates a sorted instruction list $\mathcal{C}({}_{\varkappa}\hat{\mathcal{S}},\ {}_{\varkappa}\bar{\mathcal{S}})$ for the *Covariant ICP* module. Segment correspondence is obtained using a two-way nearest-neighbours approach, refined using search masks according to agent and objects pose uncertainties $\mathbf{\Sigma_x}$.

d) Covariant ICP: The module evaluates the segment sorted pairs list and computes, if found, the according rigid transformations. In comparison with other approaches, the module identifies situations of symmetry (circular arc, perfect line, repetitive patterns) and includes this information by computing transform uncertainty. Thus, false data association is reduced and object merging/splitting can be evaluated.

e) LSLAM: The loop is being closed by an EKF type estimator. The objects are initially modelled as constant-acceleration dynamic with uncorrelated translation and rotation, assuming small values of process noise. However, their estimation convergence is evaluated and additional process noise is added (e.g. in situations of objects undergoing high accelerations). This way, objects that are in steady state (static, constant velocity) have reduced relative uncertainty and thus weight more in agent self-pose correction. In a nutshell, the agent learns the dynamics of the environment when its odometry is accurate. When odometry inconsistency is detected (wheel slips, time delay), noise injection in the agent state will automatically correct its pose in a probabilistic manner, using the most certain landmarks state as reference. However, if all objects are assumed to be dynamic, agent noise injection implies degradation of the entire state estimation, being feasible only for short-term inconsistencies.

III. Experimental results

Experiments have been conducted using the Gazebo simulation environment under Robotics Operating System (ROS). The agent is a Pioneer-P3DX mobile robot equipped with a Hoyuko laser range sensor. The agent and sensor dynamics and noise are modelled and simulated accordingly: constrained agent velocities ($a_{max} \sim 0.4m/s^2$); Gaussian noise in sensor range ($\sigma_\phi \sim 10^{-2}m$) and agent velocity ($\sigma_v \sim |v| \cdot 10^{-2}m/s$). As loop skips or low frequencies of the filter will only increase the estimation uncertainty, the framework frequency is set to 8Hz (ROS publishes laser and odometry data unsynchronised at 10Hz). For all the experiments, the agent is tele-operated.

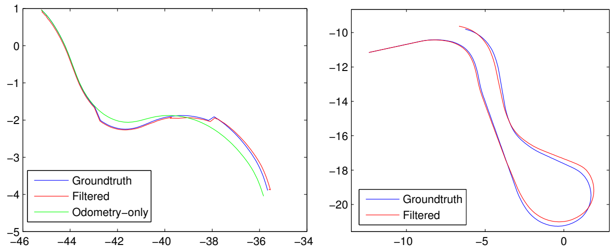

Fig. 3: Self-pose correction in dynamic (*left*) and static (*right*) environments ($m \times m$)

Object correspondences: Even though the framework provides short-term memory differential mapping, higher-level features such as object correspondences are being extracted. Figure 1 illustrates capabilities of the framework to successfully track identified objects in non-trivial scenarios.

Self-pose correction: As described in Section 2, the pose of the agent is expected to be corrected with high degrees of accuracy as long as parts of the environment are in steady state. Figure 3 (*left*) presents the filtered trajectory of the agent in dynamic environments when it undergoes short-term deviations from the motion model. Even though the framework has been designed for agents with relatively accurate odometry and a dynamic word assumption, it can as well be operated assuming a static environment. Figure 3 (*right*) presents the filtered agent trajectory, assuming a static world and feeding into the framework fixed angular and linear velocity readings set to 0 (odometry impairment).

IV. Conclusions

Agent self-pose correction in dynamic environments is still a weakly addressed problem within the robotics community. The presented framework has proven to extract sufficient information from partially steady-state dynamic environments, even though low constraint models of the environment and agent are assumed. Following such a general approach, developments towards static objects detection (e.g. using static map information) should improve the overall estimation and tracking, providing a complete solution for SLAM with dynamic descriptors and potentially broadening the capabilities of robotics tasks that make use of local maps.

References

[1] S. Choudhary, A. J. B. Trevor, H. I. Christensen, and F. Dellaert. SLAM with object discovery, modeling and mapping. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 1018–1025, 2014.

[2] Robert MacLachlan. Tracking of moving objects from a moving vehicle using a scanning laser rangefinder. Technical report, Carnegie Mellon University, 2005.

[3] Luis Montesano. *Detection and tracking of moving objects from a mobile platform. Application to navigation and multi-robot localization*. PhD thesis, Universidad de Zaragoza, 2006.

[4] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press , 2005.