MoDSeM: Modular Framework for Distributed Semantic Mapping*

Gonçalo S. Martins, João Filipe Ferreira, David Portugal and Micael S. Couceiro

***Abstract*— This paper presents MoDSeM, a novel software framework for spatial perception supporting teams of robots. MoDSeM aims to provide a semantic mapping approach able to represent all spatial information perceived in autonomous missions involving teams of field robots, and to formalize the development of perception software, promoting the development of reusable modules that can fit varied team constitutions. Preliminary experiments took place in simulation, using a 100x100x100m simulated map to demonstrate our work-in-progress prototype's ability to receive, store and retrieve spatial information. Results show the appropriateness of ROS and OpenVDB as back-ends for supporting the prototype, achieving promising performance in all aspects of the task and supporting future developments.**

I. Introduction

The **Mo**dular Framework for **D**istributed **S**emantic **M**apping (MoDSeM) aims to provide a semantic mapping approach able to represent all spatial information perceived in autonomous missions involving teams of field robots, such as those operating in precision forestry missions, aggregating the knowledge of all agents into a unified, cohesive representation. It also aims to formalize and normalize the development of new perception software, promoting the implementation of modular and reusable software that can be easily swapped according to the sensory abilities of each individual platform. This text presents an overview of MoDSeM and of some preliminary experiments using a work-in-progress implementation that evaluate the main design choices in a simulated forestry environment.

This article is structured as follows: the remainder of this section focuses on highlighting the paper's contributions, Section II presents the MoDSeM architecture, Section III presents our preliminary experiments and results, which are discussed in Section IV. Lastly, Section V presents our conclusions and future work.

A. Contribution and Related Work

MoDSeM's originality and contribution to the field lies in its focus on improving the technology readiness level (TRL) [1] of cooperative perception techniques, and on enabling them to operate in a coordinated, flexible and seamless manner. In fact, while works in perception are abundant, including probabilistic approaches [2] and works on several sub-problems of field robotics, including tree detection [3], crop/weed discrimination [4] or detection of plant disease [5]; very few of these techniques are available as easily-reusable software packages. These software packages, which include for instance mapping and localization techniques [6][1],[7][2], among others, constitute the most accessible way of testing perception techniques in the field, in conditions as close to real operation as possible. However, these packages represent only a small subset of the substantial body of work in perception, and are traditionally quite behind the state of the art. MoDSeM aims to tackle this issue by providing the means to integrate the output of these techniques, to make them usable by heterogeneous teams of robots, and by providing guidelines and formalisms for the development and integration of Perception Modules.

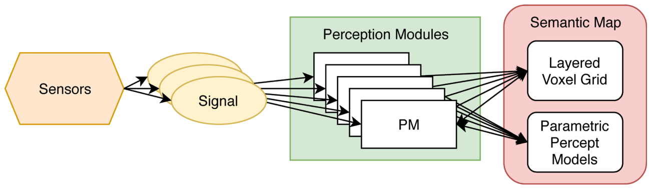

Figure 1: An overview of MoDSeM. Sensors produce signals, which are passed to independent perception modules. Percepts obtained by these modules are aggregated in a Semantic Map, containing layers for different kinds of information.

Software frameworks for robots have been developed for generic robots, such as ROS [8], YARP [9] or GenoM3 [10], and also specifically for agriculture and forestry robots [11]. These frameworks focus on improving software portability, and introduce some standards[3] on the basic common features and assumptions of the various modules. However, these frameworks do not tackle the particular issues of perception systems, such as achieving a common representation of the world with varying sensor input, or the storage and retrieval of this information, both current and in the past. Past efforts also do not define a development methodology to produce portable perception software, one of the main long-term goals of MoDSeM. To the best of our knowledge, MoDSeM is the first attempt at such a system and methodology applied directly to the problem of perception.

*This work was supported by the Seguranças robóTicos coOPerativos (STOP, ref. CENTRO-01-0247-FEDER-017562), the Safety, Exploration and Maintenance of Forests with Ecological Robotics (SEMFIRE, ref. CENTRO-01-0247-FEDER-03269) and the Centre of Operations for Rethinking Engineering (CORE, ref. CENTRO-01-0247-FEDER-037082) research projects co-funded by the "Agência Nacional de Inovação" within the Portugal2020 programme.

Gonçalo S. Martins and Micael S. Couceiro are with Ingeniarius, Rua Coronel Veiga Simão, Edifício B CTCV, 3025-307 Coimbra, Portugal, (e-mail:{gondsm, micael}@ingeniarius.pt).

João Filipe Ferreira and David Portugal are with the Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal (e-mail: {jfilipe, davidbsp}@isr.uc.pt).

João Filipe Ferreira is also with Nottingham Trent University, Computational Neuroscience and Cognitive Robotics Laboratory, School of Science and Technology, Nottingham, United Kingdom (e-mail: joao.ferreira@ntu.ac.uk).

[1] https://github.com/OctoMap/octomap

[2] https://github.com/introlab/rtabmap

[3] *E.g.* ROS's REP 103: http://www.ros.org/reps/rep-0103.html

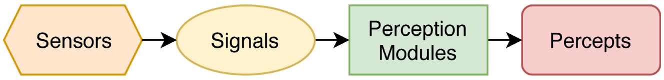

Figure 2: Traditional perception techniques implement a linear flow from sensors to percepts; signals are processed and percepts are output.

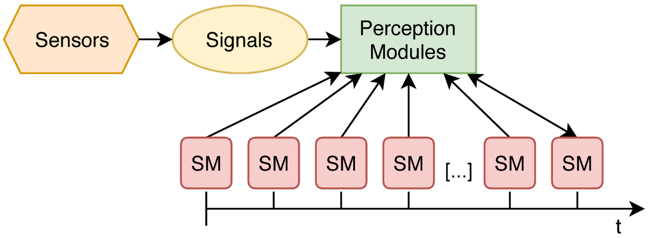

Figure 3: MoDSeM's non-linear perception pipeline: Perception Modules (PMs) are allowed to access previous versions of the Semantic Map (SM).

II. MoDSeM

A. Overview

The framework is split into three main blocks (Fig. 1): the **Sensors**, which provide raw signals; the **Perception Modules** (PMs) which take these signals and produce percepts; the **Semantic Map** (SM), containing a unified view of the state of the workspace/world which can be used by any agent in the team to make decisions or to coordinate with others.

Each PM is expected to be decoupled from other modules, depending only on the available sensors and on the SM itself, ensuring that they become interchangeable, plug-and-play elements of the system, able to be swapped at will, depending on the computational power and available sensors on each robot. This allows for the employment of PMs in different systems without the need to re-design the global representation.

The semantic map works as a global output of the system, split into two components: the Layered Voxel Grid (LVG) and the Parametric Percept Models (PPM). Each layer of the LVG is itself a voxel grid containing information on a specific aspect of the world, such as occupancy or task-relevance (e.g. the presence of certain kinds of vegetation). The combination of these layers represents the state of the world as perceived by the robot team; individually, they provide insight that may be relevant on a particular aspect of the mission. PMs can contribute to different layers of the LVG, *e.g.* with a people detector contributing to a people occupancy layer and a mapping technique contributing to an occupancy layer. The PPM complements the LVG, representing entities without volume, *e.g.* robot poses or human joint configurations.

MoDSeM aims to introduce non-linearity in the traditional data flow used in perception (Figs. 2 and 3), allowing PMs to access current and past percepts through the SM: PMs are allowed to use the SM and previous version of it as input. Indeed, some PMs are expected to use solely the SM as input; *e.g.* a traversability detector could estimate the traversability of the map using only occupancy and vegetation information.

Thus, a history of SMs is kept during operation, which could quickly make its storage infeasible. This can be mitigated, for instance, by storing the successive differences in time between the PMs as they are generated, as done in

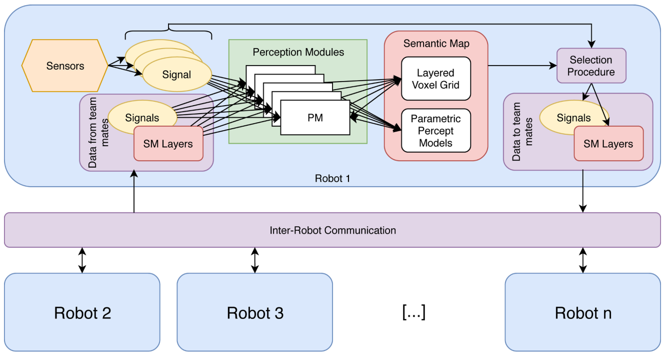

Figure 4: An overview of a robot team operating with MoDSeM. Each team member has its own sensors, perception modules and semantic map. These are shared with the rest of the team as needed, with each robot being able to receive signals and SM layers from other robots, fusing them to achieve a unified SM.

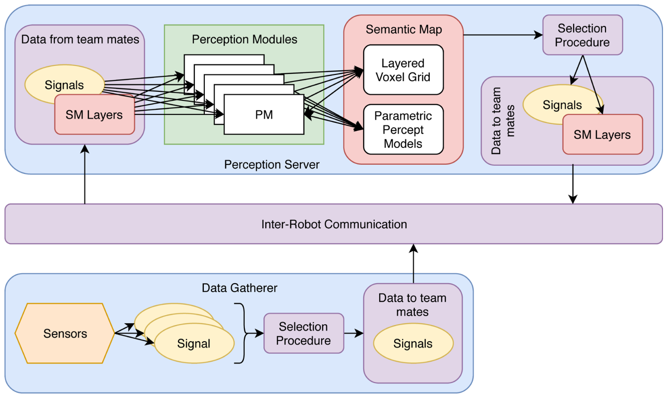

Figure 5: Different topologies for multi-robot perception using MoDSeM. Top: a perception server, which receives information and SM layers from the team and executes the most computationally expensive perception modules. Bottom: a data gatherer agent, which collects and sends data for processing in other agents.

video compression algorithms and source control systems, or by intelligently choosing which snapshots of the SM should be saved, using information-theoretic techniques.

B. Instantiation Examples

Traditionally, multi-robot perception is achieved in one of two ways: by propagating raw signals from each robot to a centralized perception server, which then replies with percepts; or by endowing each team member with perceptive abilities, as well as the ability to decide when percepts should be propagated among the team. Both of these approaches are valid in their own conditions, and it is important that MoDSeM support all of these perceptual topologies, as they allow for greater implementation flexibility.

Distributed perception is the appropriate technique when bandwidth is limited, when robots have heterogeneous needs and capabilities, and when each individual robot can be endowed with perception abilities that fit its needs. Fig. 4 illustrates an overview of MoDSeM implemented on a robotic team performing distributed perception, with each team member implementing the full architecture. In this case, each agent contains additional specialized PMs that are used to fuse information received from other agents, to achieve consensus, and selection procedures, which select information for sharing with other agents. Specific PMs in each robot can then fuse

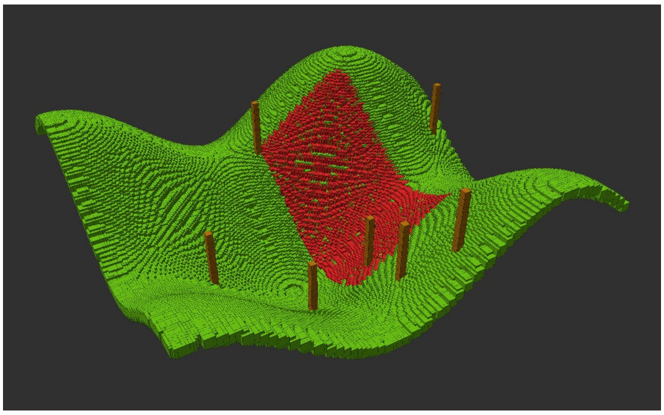

Figure 6: An illustration of the semantic map used in these tests. Green voxels represent the terrain, brown voxels represent trees and red voxels represent dry shrubbery.

these representations, achieving consensus in representation and allowing all robots to plan with the same information. These can be implemented using the same formalism as regular PMs, with no necessary particularity, and would only depend on the SM itself to function.

Centralized perception can be useful when the robots in the team carry much less processing power than the necessary perception modules and when the communication infrastructure is always available and can support the necessary bandwidth. Fig.5 illustrates the usage of MoDSeM on a centralized topology, with a data gatherer collecting and selecting signals, which are then sent to a centralized server for processing and dissemination.

Other topologies can be achieved with the framework by mixing-and-matching the necessary components, such as PMs and sensors and their configurations, to achieve different use cases. For instance, a hybrid approach can be used with heterogeneous teams, when for example one of the robots is significantly more powerful in computational terms than other team members, which in turn can therefore unload part of their perceptual load to this team mate, while still executing basic PMs.

III. Experiments and Results

Tests were conducted with the goal of assessing the appropriateness of OpenVDB [12] and ROS[4] as back-end modules for MoDSeM, exploring two main functions: data insertion into the semantic map, and data retrieval from the semantic map. These constitute the two main operations that the semantic map server is expected to perform during runtime, and should operate efficiently enough to allow real time operation. To this end, a 100-by-100-by-100 meter map was generated in simulation, at a resolution of 5cm/voxel (Fig. 6[5]), containing three semantic layers with a total of 18 million voxels. The map was sent piece-by-piece over ROS to the semantic map server, operating on OpenVDB, simulating the operation of a mapping node that advances through the terrain and iteratively updates the global map. At each update, the server was asked for the retrieval of the same portion of the map, testing its ability to deliver data. The whole experimental procedure took place on a computer running Ubuntu 16.04, equipped with an Intel Core i7-7700 and 16GiB of RAM.

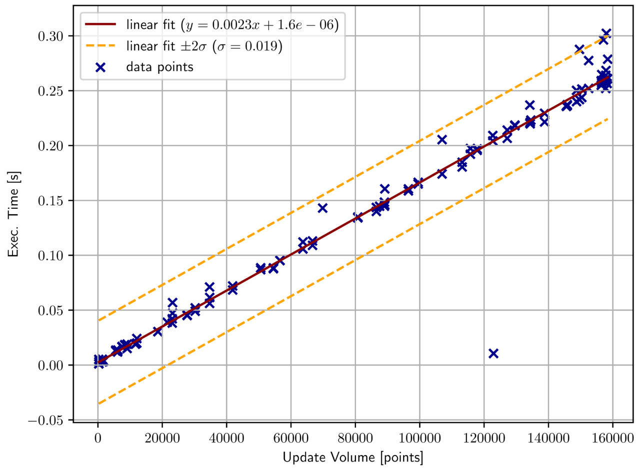

Figure 7: Time needed to update the map as a function of the size of the update, in occupied points. The clear outlier corresponds to the very first insertion of the map, wherein the received grid itself is used.

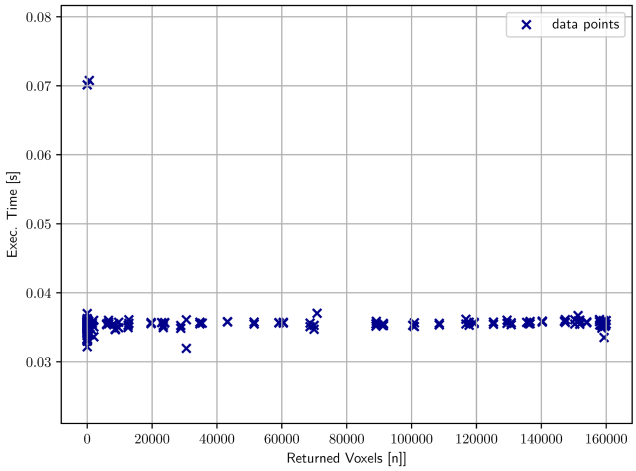

Figure 8: Time taken to retrieve a sub-map (or sub-grid) as a function of the number of voxels contained in the sub-map.

Fig. 7 illustrates the time it took to update the map as a function of the size, in occupied points, of the respective update. A linear trend is observable in the data: the update time of the map is predictable given the size of the update, and can be accounted for. In the worst-case scenario, the update procedure took around 0.3 seconds, for an update of 160,000 points.

Fig. 8 illustrates the performance of the semantic map server when retrieving a subsection of the map. We can observe no undesirable relationship between the voxel structure of the original grid and the time it takes to retrieve the grid; for a constant sub-grid volume, the retrieval time is almost constant.

Fig. 9 illustrates the semantic map server's usage of memory as the experiment progresses. We can observe that memory usage grows linearly with the size of the updates that are received, not with the mapped volume itself. This means that the map is capable of storing information independently of the

[4] http://www.ros.org

[5] Aliasing artifacts are caused by downsampling applied for visualization. It is not possible to represent the map's near-20-million voxels on rviz.

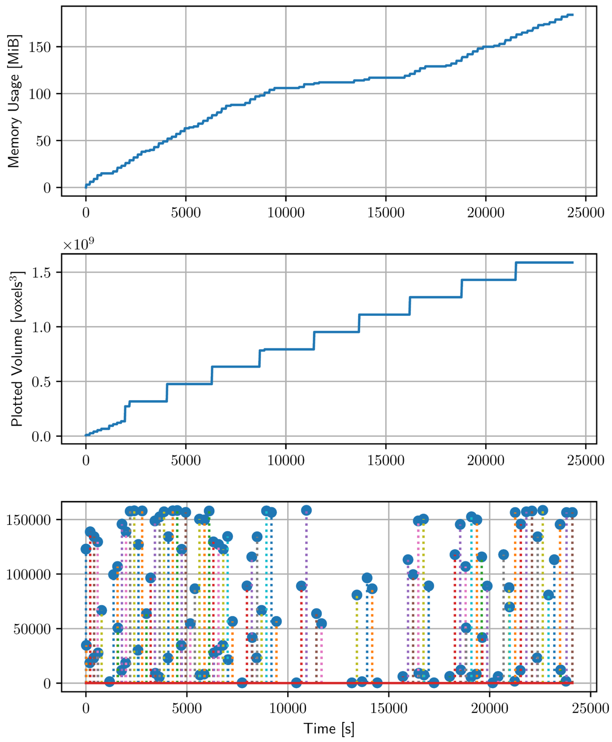

Figure 9: Evolution of memory usage as updates are received by the server. (Top) represents the total memory usage of the map; (middle) illustrates the volume, in cubic voxels, of the map; (bottom) shows the update times of non-zero updates, as well as their size in points.

volume mapped, which is one of the greatest advantages of tree-based maps such as OpenVDB or Octomap [6].

IV. Discussion

Generally, the current results are promising. Map update and retrieval speeds are fast enough for our application: updates up to 160,000 voxels, equivalent to a completely full $3m^3$ volume, can be processed at 3 to 4Hz. Given that a time complexity below *O(n)* was unlikely , this is a positive result: the update time of the map is easily predictable given the size of the update, and measures can be taken to account for it.

As seen in Section III, system performance is acceptable for the worst-case scenario. An update as large as those described therein is a relatively unlikely event; it may correspond to an update to the map produced by a mapping node or eventually to a bulk update from another perception node which has produced a large calculation. It is unlikely that such updates would be produced frequently, or at a high enough frequency to overload the server. This work demonstrates that basic functionality is possible, and that MoDSeM's future development should involve OpenVDB and ROS: they seem able to support the operation of the semantic map server and can provide a stable framework for future development.

V. Conclusion and Future Work

This paper presents the design and the ongoing development of MoDSeM, a software framework for spatial perception supporting teams of robots. Preliminary experiments confirm the appropriateness of our design choices.

We present only a preliminary study of the functionality that is being designed and implemented. It will now be extended in several ways, namely to further evaluate the framework's limitations, and to apply it in real use cases in several current research projects. Additional testing will be conducted in semi-realistic scenarios, involving several robots, to develop the inter-robot communication and data transmission facilities that will propagate SM layers across team mates. MoDSeM will then be implemented in teams of patrolling robots[6] for surveillance and inspection, allowing the team of robots to synchronize and fuse perceptual information, promoting coordinated action. In later follow-up work, MoDSeM will also support the perceptual mechanisms of heterogeneous teams of robots for automated forestry tasks [13][7].

References

[1] J. C. Mankins, "Technology Readiness Levels," *White Pap. April*, vol. 6, no. 2, p. 5, 1995.

[2] J. F. Ferreira and J. Dias, *Probabilistic Approaches for Robotic Perception*. Springer International Publishing, 2014.

[3] T. Hellström, A. Ostovar, T. Hellström, and A. Ostovar, "Detection of Trees Based on Quality Guided Image Segmentation," in *Second International Conference on Robotics and associated High-technologies and Equipment for Agriculture and forestry (RHEA-2014)*, 2014.

[4] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss, "UAV-Based Crop and Weed Classification for Smart Farming," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3024–3031.

[5] R. Oberti, M. Marchi, P. Tirelli, A. Calcante, M. Iriti, and A. N. Borghese, "Automatic detection of powdery mildew on grapevine leaves by image analysis : Optimal view-angle range to increase the sensitivity," *Comput. Electron. Agric.*, vol. 104, pp. 1–8, 2014.

[6] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[7] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. F. Robot.*, no. May, 2018.

[8] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, "ROS: an Open-Source Robot Operating System," in *ICRA workshop on open source software*, 2009.

[9] P. Fitzpatrick, E. Ceseracciu, D. E. Domenichelli, A. Paikan, G. Metta, and L. Natale, "A middle way for robotics middleware," *J. Softw. Eng. Robot.*, vol. 5, no. September, pp. 42–49, 2014.

[10] A. Mallet, C. Pasteur, M. Herrb, S. Lemaignan, and F. Ingrand, "GenoM3 : Building middleware-independent robotic components," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, no. May 2014.

[11] T. Hellström and O. Ringdahl, "A software framework for agricultural and forestry robotics," in *International Conference on Robotics and associated High-technologies and Equipment for agriculture*, 2012, pp. 171–176.

[12] K. Museth, "VDB: high-resolution sparse volumes with dynamic topography," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–22, 2013.

[13] M. Couceiro, D. Portugal, J. F. Ferreira, and R. P. Rocha, "SEMFIRE: Towards a new generation of forestry maintenance multi-robot systems," in *IEEE/SICE International Symposium on System Integrations (SII 2019)*, 2019.

[6] http://stop.ingeniarius.pt/

[7] http://semfire.ingeniarius.pt/