A framework to manage environment models in multi-robot teams

Pierrick Koch, Simon Lacroix

A framework to manage environment models in multi-robot teams

(extended abstract)

Pierrick Koch[1,2] and Simon Lacroix[1,3]

I. INTRODUCTION

Environment models are at the heart for the autonomy of mobile robots, as they constitute the main information on which planning processes rely to select and configure the tasks to achieve to fulfill a given mission. Their *consistency*, in the sense of their fidelity to the reality of the information they represent, is therefore of utmost importance, as any discrepancy or error will eventually lead to wrong decisions. When it comes to teams of robots operating in the same environment, sharing environment models is a pre-requisite to cooperation: whatever the mission to achieve (*e.g.* exploration, surveillance), the robots must indeed have a common understanding of the situation at hand to define consistent and efficient cooperation schemes.

The robotics literature naturally abounds with contributions on the *building* of various kinds of environment models from the data gathered by robots. Similarly, a vast amount of work has been devoted to robot localisation, which is in particular required to ensure the *spatial consistency* of the built models, either in mono or multirobot contexts. Yet much fewer contributions can be found on the *managing* of environment models within a team of robots, which consists in *(i)* structuring them so as to properly serve the processes that exploit them, *(ii)* making sure that the most precise information is extracted from the available data, in spite of the dynamics of the environment and of the sensors and positioning errors.

This paper presents a *framework* dedicated to the management of environment models among mobile robots. The considered context is large scale outdoor environments, of which initial models may or may not be known, and in which a team of robots of various kinds jointly achieve long duration missions that mainly relate to environment perception, *e.g.* in search and rescue or environment monitoring contexts (tasks such as logisitics and transportation are not considered, even though the framework could easily be adapted to such missions). The environment is supposed to be non-networked, in the sense that the robots can not rely on a ubiquituous, large bandwith communication infrastructure that would allow them to benefit from remote powerful storage and computation servers. Yet the robots are of course endowed with communication abilities, that are however constrained by the environment in range and bandwidth: communication activities have to be actively controlled by the team, and the framework must offer means for this purpose.

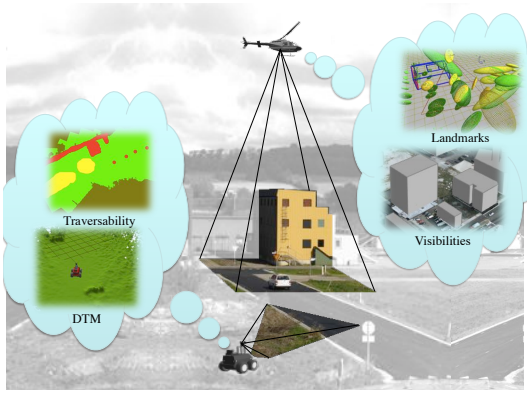

Fig. 1. An AAV and an AGV jointly operate in the same environment. For this purpose, they exploit a series of environment-related information to plan, coordinate and control their activities (*e.g.* environment traversability, visibility, communication and localisation constraints, ...). These information are encoded into environment models, that integrate a variety of sources. How to structure these various models, so as to ensure their consistency within the team? How to enable their sharing between the robots?

Defining such a framework requires to face the usual challenges with which environment modeling and robot localisation algorithms have to cope: uncomplete and noisy data, variety of information sources and quality (including initial environment models), environment dynamics... These points are crucial in the design of the modeling and localisation algorithms, and have driven more than two decades of research in robotics, that mostly relate to uncertain data fusion. Yet, the maintenance of spatial and temporal consistency of the models within the team calls for additional concerns, such as the memorization of the gathered data, *e.g.* so as to rebuild a model corrupted by wrongly positioned data when past localisation information are updated. The heterogeneity of the robots also impacts the management of the models: for instance some landmarks can be exploited to localize a given kind of robots and not others that have no mean to perceive them, or the traversability information are robot-dependent. Finally, the communication constraints impose a distributed solution to manage the various models.

To takle these challenges, the proposed framework im-

[1]CNRS, LAAS, 7 av du colonel Roche, F-31400 Toulouse, France; {pierrick.koch, simon.lacroix} at laas.fr
[2]Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France
[3]Univ de Toulouse, LAAS, F-31400 Toulouse, France

plements two basic principles: *purposiveness* of the models, that are defined and built according to the processes that exploit them, and *economy of means*, be they related to computations, storage and communications.

Outline: The next section reviews some contributions of the litterature that tackle some aspects of the management of environment models within robot teams. Section III then presents the way the *purposiveness* principle is applied, after an analysis of the relations between environment models and the various processes that exploit them (mainly decision and planning, but also localisation). Section IV then depicts the way environment models and data are structured in a dedicated database, which allows their dynamic management. Illustrative preliminary results are presented in section V, and a discussion concludes the paper.

II. Related work

Most of the multirobot contributions on the building of environment models come to localisation, either of the robots themselves or of targets detected in the environment, for which the distribution of sources has fostered theoretical analyses [1], [2] that have led to practical implementations [3]. This is not surprising, as on the one hand localisation is at the core of the building of environment models, and on the other hand multiple robots yields new means to ensure localization – *e.g.* by defining additional loop-closures in SLAM approaches [4]. Besides localisation, environment related information being at the core of search or exploration missions, numerous contributions deal with the fusion of such information in a multi-robot context (*e.g.* [5], [6]).

Much less contributions can be found on the managing of environment models. By proposing a "stream-based knowledge processing middleware" DyKnow[1] rather focuses on symbolic information propagation within a system, handling in particular their time properties. When it comes to environment models, some contributions exploit "torrent-like" tools developed for web-based distributed map management [7], which comes to manage a distributed database over a cloud infrastructure.

III. Environment models and decisional processes

Decisional processes rely on both environment and action models, actions being environment observations (perception), robot motions and inter-robot communications. No single environment model can represented the whole spectra of information required to plan theses processes, and hence the environment models are structured into a set of *layered representations*, each representation being dedicated to plan a given action:

- Models that represent the geometry of the terrain are required to plan motions. A digital elevation model (DEM) is required for local mobility, achieved by a short term path planner that must be fast and real-time; a more abstract description of the world is needed for higher level of decision, as the mission planner, demanding lower resolution and less time constraints (traversability model);
- 3D models are required to evaluate the communication and visibility of an area from a given position
- *Utility* models encode the interest of observing areas, *e.g.* the probability that it contains an event to monitor (orthoimage);
- Dedicated models are required for localization, exhibiting characteristic elements in the environment (landmarks)
- ...

Yet these environment related models must be tailored to the considered action models, which may vary from one robot to the other. Following a "clear separation of concerns" principle, these various models are structured within a library that exhibits the various kind of models, that act as a server for the decisional processes that must evaluate the outcome of planned actions (figure 2).

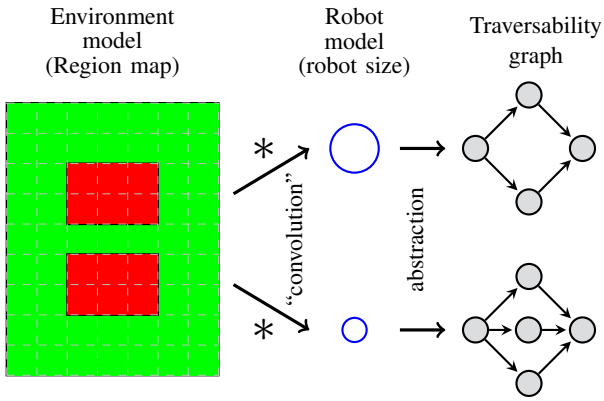

Fig. 2. Illustration of the model abstraction process, applied here to motion actions: by convolving an environment-centric (robot independant) digital terrain map with two different robot motions models, two different accessibility graphs are produced.

IV. Distributed environment models

So as to be able to handle relocations and to serve the decisional client processes with the adequate environment information they require, we propose to use tiles to keep track of past modelled areas. This is achieved by subdividing space into a set of smaller maps, allowing to cache data, dump and load them as the robot moves to different areas.

A. Pile of tiles

A tile is composed by different layers, for each cell, we are currently storing the lowest and highest elevations perceived, the elevation mean and variance, the number of points it contains, and the timestamp of the last update.

The current implementation merges incoming point clouds in a map of 9 tiles ($3 * 3$). When the robot moves out of the central tile, we dump the ones out of the map and load existing if any, using a simple hysteresis threshold. For local path planning purposes, we build maps of 0.1 meter

[1] http://www.ida.liu.se/divisions/aiics/projects/dyknow.en.shtml

resolution, of size $40 \times 40m^2$: this size is selected so as to make sure each tile is spatially consistent, making the hypothesis that the robot localization drift is not exceeding the cell resolution over the tile size (we are using inertial-visual SLAM [9]).

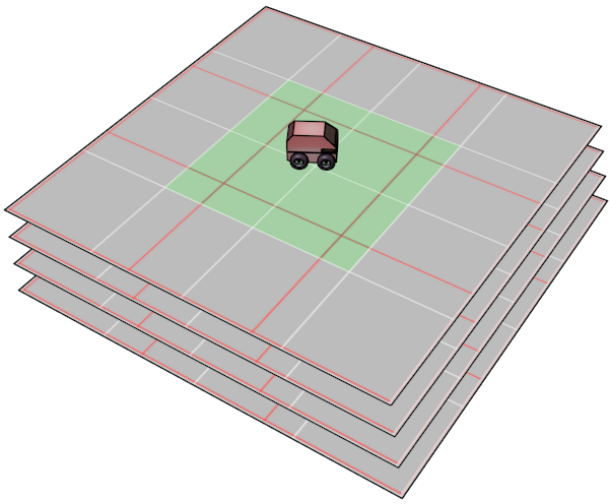

Fig. 3. Current map of $3 * 3$ tiles with their layers. You can see the tiles delimitation in red, and in green the central "hysteresis" robot area (best view with colors).

Each tile being spatially and temporally consistent, and can be treated as an independent dataset containing its own relative frame and the transformation to the global frame. This hierarchy [10] gives the possibility to reshape the overall model as the robot position is updated – *e.g.* after a loop closure detection and correction by the SLAM algorithms. The overall model is generated by composed tiles, hence can quickly be re-arranged.

Space subdivision is also a key for sharing data between robots, sending chunk of information on demand, and registering them in another robot frame using remote synchronization methods.

B. Position referencing

In order to share models in a multi-robot scenario, it is first necessary to rely on standard for sharing frames in space and time. We are using the Universal Transverse Mercator (UTM) coordinate system to refer the position of each of the tiles, to be consistent with GPS position estimations – and with existing maps.

A tile is one file, a multi-layer GeoTiff/Float32 saved using the Geospatial Data Abstraction Library (GDAL). We built a generic library wrapping GDAL I/O using modern C++11 STL containers for in-memory storage [3].

We keep an history of past tiles for the same area and use registration based on "stable" cells by computing the probability of the dynamic of each cell. It is when tiles overlap that we will handle the issues, by properly "merging" tiles, after having registered them (either by directly correlating them [11], or matching landmark maps associated to the tiles).

Furthermore, keeping only one frame of reference is not sufficient. It is desirable to have a model that is still fixed as the robot moves forward and when loop closure appears. Hence the track of local reference in the global model should be kept and transformed to local chunk of data to prevent corruption. Doing so while sharing models means that tiles with reference to the global frame and information about its potential error (covariance) must be memorized.

V. Illustration

The final paper will exhibit the behavior of the overall infrastructure in a multi-robot exploration scenario, showing its ability to maintain a global consistent environment models database over time, and analysing in particular the required communication bandwidth between the robots.

[2]For 9 tiles of $40 * 40m$, the memory footprint is 33 MB

[3]http://trac.laas.fr/git/gdalwrap

References

[1] E. W. Nettleton and H. F. Durrant-Whyte, "Delayed and asequent data in decentralized sensing networks," in *Proc. SPIE*, vol. 4571, 2001, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1117/12.444148

[2] A. Makarenko, A. Brooks, T. Kaupp, H. Durrant-Whyte, and F. Dellaert, "Decentralised data fusion: A graphical model approach," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, July 2009, pp. 545–554.

[3] D. T. Cole, P. Thompson, A. H. Göktogan, and S. Sukkarieh, "System development and demonstration of a cooperative UAV team for mapping and tracking," *International Journal of Field Robotics*, vol. 29, no. 11, pp. 1371–1399, 2010.

[4] T. Vidal-Calleja, C. Berger, J. Solà, and S. Lacroix, "Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 654–674, Sept. 2011.

[5] G. A. Hollinger, S. Yerramalli, S. Singh, U. Mitra, and G. S. Sukhatme, "Distributed Data Fusion for Multirobot Search," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 55–66, Feb. 2015.

[6] A. Khan, E. Yanmaz, and B. Rinner, "Information Merging in Multi-UAV Cooperative Search," in *IEEE International Conference on Robotics and Automation*, 2014.

[7] T. Cieslewski, S. Lynen, M. Dymczyk, S. Magnenat, and R. Siegwart, "Map API - Scalable Decentralized Map Building for Robots," in *IEEE International Conference on Robotics and Automation*, 2015.

[8] S. J. Julier and J. K. Uhlmann, "Real-time distributed map building in large environments," in *Proc. SPIE*, vol. 4196, 2000, pp. 317–328. [Online]. Available: http://dx.doi.org/10.1117/12.403735

[9] C. Roussillon, A. Gonzalez, J. Solà, , J.-M. Codol, N. Mansard, S. Lacroix, and M. Devy, "Rt-slam: a generic and real-time slam architecture," in *International Conference on Vision Systems, Sophia Antopolis (France)*, Sept. 2011. [Online]. Available: http://homepages.laas.fr/simon/publis/ROUSSILLON-ICVS-2011.pdf

[10] P. Beeson, J. Modayil, and B. Kuipers, "Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy," *The International Journal of Robotics Research April 2010 vol. 29 no. 4 428-459*, pp. 428–459, 2010.

[11] B.-V. Pham, A. Maligo, and S. Lacroix, "Absolute map-based localization for a planetary rover," in *12th ESA Workshop on Advanced Space Technologies for Robotics and Automation, Noordwijk (The Netherlands)*, May 2013.

[12] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 3185–3192.

[13] A. Kassir, R. Fitch, and S. Sukkarieh, "Communication-aware information gathering with dynamic information flow," *The International Journal of Robotics Research*, vol. 34, pp. 173–200, 2015. [Online]. Available: http://ijr.sagepub.com/content/34/2/173.abstract