



# Local Map Extrapolation in Dynamic Environments

Romain Drouilly, Panagiotis Papadakis, Patrick Rives, Benoit Morisset

► **To cite this version:**

Romain Drouilly, Panagiotis Papadakis, Patrick Rives, Benoit Morisset. Local Map Extrapolation in Dynamic Environments. IEEE International Conference on Systems, Man, and Cybernetics, Oct 2014, San Diego, United States. 2014. <hal-01075285>

**HAL Id: hal-01075285**

**<https://hal.inria.fr/hal-01075285>**

Submitted on 17 Oct 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Local Map Extrapolation in Dynamic Environments

Romain Drouilly<sup>1,2</sup>, Panagiotis Papadakis<sup>1</sup>, Patrick Rives<sup>1</sup>, Benoit Morisset<sup>2</sup>

<sup>1</sup>Team LAGADIC, INRIA Méditerranée, Sophia-Antipolis, France

email: {romain.drouilly, panagiotis.papadakis, patrick.rives}@inria.fr

<sup>2</sup>ECA Robotics, email: bmo@eca.fr

**Abstract**—We present a generative approach to perform robot mapping that is based on an intelligent integration of static and dynamic entity classes within an environment, in order to extrapolate map information at various resolutions. Our framework differentiates from the conventional standpoint where different mapping levels are overlaid on one another, by fusing information from different mapping levels that allows us to infer new information within partially mapped environments. Towards this goal, we develop a class-dependent map extrapolation function that captures the discriminative relation between an environment entity and the mapping procedure. We illustrate the advantages in using heterogeneous contextual information when mapping an environment using a prototype implementation of our approach on an indoor robot platform, giving very promising results.

## I. INTRODUCTION

There is a growing trend in robotics to consider contextual information as a means to augment the perceptual boundaries of contemporary robots. This is largely motivated by the increasing integration of robots into the real world wherein humans interact and operate by exploiting explicit as well as implicit knowledge. Likewise, integration of a priori knowledge into robotic skills is a required but not sufficient condition for deploying robots into increasingly complex environments, that should further be equipped with special skills in order to account for incomplete information. Among these skills, the ability to map the environment has for long been considered as a prerequisite for developing intelligent path-planning behaviours with most research efforts focusing on building maps that capture as precisely as possible the underlying environment structure. However, mapping precision deteriorates within changing environments due to dynamic events that cause occlusions or impede access to certain areas mainly due to human-incurred events. In turn, this results in maps that are inappropriate or insufficiently complete for robot operation since mapping techniques [1] typically discard changes as outliers and instead seek to preserve only what appears to be spatially consistent over time.

When mapping the environment, the conventional approach is to progressively abstract information into various levels from the *metric* (2D or 3D) and *topological* [2] to the *semantic* [3], [4] and *social* layers [5] that are organized into a hierarchy where information is processed in a bottom-up fashion. We argue that a top-down analysis can assist in augmenting the information that has already been acquired, by performing extrapolation that is guided by a semantic interpretation of both its static and dynamic characteristics.

In the domain of robotic mapping, the use of cues that are related to dynamic events or static scene semantics has so far been exploited separately for each case. This is mainly due to

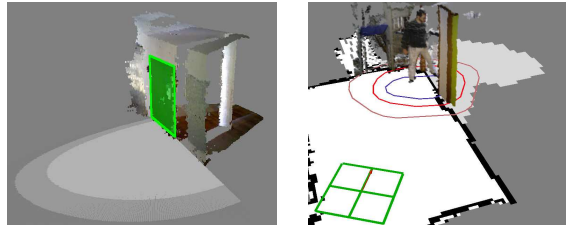


Fig. 1. Examples of map extrapolation by scene context interpretation.

the fact that mapping techniques aim at filtering out all non-static cues from the extracted environment map in an effort to derive a representation that is invariant to change and finally perform a labelling of the corresponding areas.

There is a rather sparse number of earlier works that partly address the problem of inference at the level of robotic mapping through the use of static semantic contextual information. In a preliminary account, Calisi et al. [6] presented a pilot study showing the advantages of using contextual information of the environment and internal robot states in basic navigation and mapping tasks for urban search and rescue scenarios. In the domain of structured environments, Callaghan et al. [7] developed a regression mechanism for space occupancy through the use of a Gaussian process that was trained on samples taken from earlier constructed maps and a non-stationary covariance function that adapted to the neighbourhood of query points. Their approach alleviated uncertainty due to occlusions or range limits as well as noisy measurements but without accounting for dynamic objects such as humans. In the top-down approach proposed by Nuchter et al. [3] scene labels are used to correct pose estimation to limit drift along trajectory but without seeking to extrapolate scene structure beyond perceptual limits. Finally, the work of Pronobis et al. [8] probably constitutes the most elaborate attempt to account for the static semantic content of an environment in robot mapping using a property-based scheme. However, inference is performed for areas that were already partially observed and in the complete absence of dynamic events.

Mapping the non-static constituents of an environment is a relatively new problem and hence there is a limited number of earlier works that could be referred. Recently, Alempijevic et al. [9] employed Gaussian processes to refine the probability of occupancy in free space by tracing human trajectories. The individual poses along the tracked trajectories were used as training points for fitting the likelihood function which was updated in parallel to the execution of a controversial path planner that favoured the mostly visited areas. Grzonka et al. [10] demonstrated the ability to build approximate environment maps by using door closure/opening events as candidate loop

closure positions and room delimiters. A conditioned dilation operation on the 2D image of a human trajectory was employed in order to emerge the coarse structure of the environment, yet the fact that human motion was tracked remotely using MOCAP suits worn during each trial limits the utility of that approach. An interesting perspective was proposed by Jiang et al. [11], wherein the hidden context behind scene structure is revealed by selective sampling of a fixed, distinctive set of human poses within empty space and learning human-object and object-object affordances. While that approach bears similarities from the point of view of using humans to interpret scene structure, it is used for the purpose of bottom-up scene labelling while we are concerned with performing map extrapolation along both directions.

Towards this goal, we propose a framework to extrapolate map information within a partially mapped indoor environment using contextual information by interpreting the static as well as dynamic semantic content of the environment (example shown in Fig. 1). We achieve this by developing a class-dependent map extrapolation function that allows us to account for both static and dynamic aspects of an environment. In this way, our approach allows a robot to significantly augment its perceptual capacity, in contrast to earlier works that either consider one of the two constituents individually or impose an explicit distinction between them. The advantages of our approach are shown through a set of concrete examples that highlight the added value of extrapolating map information instead of mapping solely the observed areas.

We organize the remainder of the paper as follows. In Section II, we present the details of our approach and solidify its advantages within a set of characteristic use-cases, in Section III we demonstrate the efficacy of our method in various examples using an indoor robot platform and finally, summarize our findings in Section IV.

## II. PROPOSED APPROACH

When mapping an environment there will always be areas that remain uncovered by a robot resulting in the construction of an incomplete map. The distinguishing factors that are responsible for this can be *static*, *quasi-static* and *dynamic* entities of an environment such as physical barriers that occlude an area of interest and events incurred by human activity. Our purpose is to specifically account for those entities when mapping an environment and in particular, for those that mostly affect the mobility of a robot. More particularly, we rely on a set of mapping functions (i.e. robotic processes)  $M_l(\cdot) : I_l \rightarrow O_l$ ,  $l = 1, 2, \dots, L$ , each of which captures a particular aspect of the environment through the analysis of a specific set of attributes, where  $I_l, O_l$  correspond to the input and output domain of the corresponding function. In the present work, the roles of the functions are determined through a set of  $L = 3$  in total increasing environment map abstraction layers enumerated as; (1) *metric*, (2) *semantic* and (3) *social* where each level of abstraction is dedicated for a certain task. The first two mapping levels have well-known, distinct roles [12], [13], namely, the metric is used for obstacle avoidance and local navigation and the semantic for task-based, intelligent path-planning while social mapping is mainly useful for allowing socially-compliant human-robot interaction [5].

For the construction of our map structure, more specifically, we follow an ego-centric approach wherein information from the different layers is integrated into locally consistent sub-maps globally connected in a graph structure, whose edges denote the reachability among the sub-maps. Each sub-map describes the environment from a particular viewpoint in the joint domain  $\mathbb{O} = O_1 \times O_2 \times O_3$ . Our ultimate objective is then to augment the situation awareness of the robot either through the addition of new sub-maps or the augmentation of the current sub-map, using information that is already available within the mapping hierarchy and is encapsulated within the semantics of the static and dynamic scene.

### A. Local and Global Map Definition

A local sub-map  $S$  constitutes a multi-layered, ego-centred representation holding the decomposition of the environment into the metric, static semantic and human social activity level. As such, it consists of several different types of information that are characteristic of each layer, however, we will limit our formal description to only those parts that are used by the proposed methodology and only selectively refer to the remaining parts whenever necessary.

We denote a local sub-map as  $S = \{P; R; U\} \in \mathbb{O}$  wherein  $P$  is the set of points sensed in the local environment,  $R$  is the set of groups of points  $\mathbf{p} \in P$  wherein each group forms an entity assigned with a semantic label and  $U$  is the set of 2D cells on a rectangular 2D occupancy grid that encode the navigational area for the robot. There are in total  $N_P$  number of points as prescribed by the sensor resolution,  $N_R$  number of detected entity regions as obtained by the semantic entity labelling of the local sub-map (see sections II-C, II-D, II-E) and  $N_U$  occupancy grid cells as defined by the resolution of the occupancy map. Each particular entity  $r \in R$  groups the corresponding points and is appointed with a vector that holds the entity's pose (3D translation  $\mathbf{t} = (t_x, t_y, t_z)^T \in \mathbb{R}^3$  and orientation  $\mathbf{o} = (\phi, \theta)^T \in [0, \pi) \times [0, 2\pi)$ ) and a set of class-dependent attributes  $\mathbf{c}$ . We denote this vector as  $\mathbf{a}_r = (\mathbf{t}^T, \mathbf{o}^T, \mathbf{c}^T)^T$ .

Local sub-maps are globally connected in a graph  $\mathcal{G} = \{V, E\} \in G$  where  $V$  is the set of sub-maps,  $E$  the set of edges that connect sub-maps and  $G$  the set of possible graphs. We distinguish between two sets of sub-maps so that we denote  $V = S_O \cup S_P$  where  $S_O$  are the set of observed sub-maps and  $S_P$  the set of predicted sub-maps. Edges correspond to relative transforms between two sub-maps. Finally, we segment the graph in *topological places* that are delimited through the existence of *static* or *quasi-static* entities. Then two sub-maps are considered belonging to the same *topological place* if they are not connected through these entities. According to this definition, elements of  $S_P$  are considered as belonging to different topological place than those of  $S_O$ .

### B. Map Extrapolation

We construct an entity-dependent *extrapolation* function that serves as a guiding model for local map extrapolation. This function propagates information either within the local sub-map hierarchy or the global graph map. We denote the extrapolating function as  $\mathcal{M}(\cdot) : \mathbb{O} \rightarrow \mathbb{O}$  and unfold our approach using a number of selected static, quasi-static and

dynamic entity classes. The general form of this function is as follows:

$$\mathcal{M}(S_i) = \delta_{ij}S'_i + (1 - \delta_{ij})S_j \quad (1)$$

where  $\delta_{ij}$  is the Kronecker delta function,  $i$  denotes the index of the current sub-map  $S_i$  where the robot is placed and  $j$  the index of a newly created sub-map  $S_j$  that could be created as a result of map extrapolation. Essentially, eq. (1) states that new information will either be inserted to the current sub-map  $S_i$  that will be updated to  $S'_i$  or a completely new sub-map  $S_j$  will be generated, depending on the entity class. In the next sections, we will develop different forms of this function according to the semantics of the particular entity class.

### C. Static entity based extrapolation

Among the various static entity classes within the environment, some are of particular interest for robot navigation as they connect different parts of the environment. Thus, they are powerful topological cues that could be used to infer the presence of navigable space within unobserved areas.

In this work, we perform map extrapolation using staircases as static entities. To detect a staircase we first filter the 3D point cloud  $P_i$  of map  $S_i$  by extracting a voxel-based representation and then obtain planar surface patches through the application of the RANSAC algorithm. If a patch normal is oriented around  $45^\circ$  from the vertical axis and has an overall surface area greater than a minimal size it is recognized as stairs. Once the staircase is detected and its position  $\mathbf{t}$  is obtained as the centroid of the patch, we set its orientation  $\mathbf{o}$  as follows. Let  $\mathbf{n}$  be the normal of the staircase plane,  $\mathbf{z}$  the vertical coordinate axis and  $\mathbf{w} = (\mathbf{n} \times \mathbf{z}) \times \mathbf{n}$ . The orientation corresponding to the ascending stairs direction is obtained as  $\mathbf{o} = \mathbf{w} \cdot \text{sgn}(\mathbf{w}^T \mathbf{z})$ .

The stairs are further characterized by the step width  $w$  and the expected height of the new floor  $h$  at the top endpoint of the staircase so the corresponding parameter vector is expressed as  $\mathbf{a}_s = (\mathbf{t}^T, \mathbf{o}^T, (h, w)^T)^T$ . Fig. 2 gives an example of staircase detection annotated with the corresponding parameter vector.

Upon detection of the staircase, a new sub-map  $S_j$  is generated by adding a node to the graph. It is connected to the current sub-map  $S_i$  by the edge  $E_{ij}$  that corresponds to the position of the expected top of the stairs expressed in the  $S_i$  reference frame.  $S_j$  is initialized by copying entity-related data of  $S_i$  that are assumed to be shared between the two sub-maps. It is denoted as  $S_j = \{P_S; R_S; \emptyset\}$  where  $P_S$  corresponds to stairs points and  $R_S$  the corresponding group.

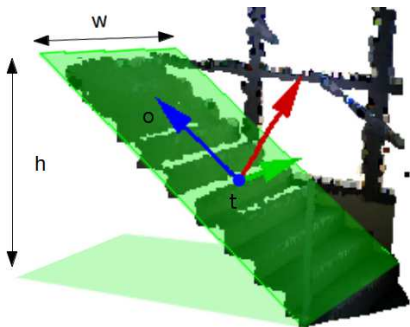


Fig. 2. Static entity class *staircase* along with its parameters.

Then it is complemented as  $S_j = \{P_S; R_S; U_j\}$  where  $U_j$  is obtained through extrapolation of free space via the rule:

$$U_j = \{\mathbf{u} \in \mathbb{R}^2 | f_s(\mathbf{u}) > p_{th}\} \quad (2)$$

where  $f_s(\cdot)$  is a bivariate normal probability density function (pdf) denoted as  $\mathcal{N}(\mathbf{0}, \Omega)$  with covariance matrix  $\Omega = I\sigma^2$ ,  $\sigma$  is proportional to the staircase step width  $w$  and  $\mathbf{u} \in U_j$  is an arbitrary point within the local occupancy map. The  $f_s(\cdot)$  function encodes the confidence for the existence of free space up to a certain distance from the origin  $\mathbf{0}$  of the new sub-map reference frame which is controlled by  $p_{th}$ .

### D. Quasi-static entity based extrapolation

Quasi-static entities are defined as entities that change occasionally and their state can have a direct effect on the mobility of a robot. The most typical example corresponds to doors due to their functional property for space delimitation, that are defined by their fixed rotational axis which limits their motion range. If a planar patch that was detected in the previous stage is orientated orthogonally to the ground it is analysed further to find potential doors. In detail, we extract salient points [14] in the 2D image of the planar patch that we subsequently cluster in two groups: (i) those that are close to the ground and are candidates to belong to the bottom edge of the door and (ii) those that are close to the door height and are candidates to belong to the top edge of the door. Then, we seek the points from the two groups that are vertically aligned and for each pair we search whether there exists another pair that resides at a distance equal to the expected width of a door. Finally, we find the rotational axis of the door by searching for the door handle. We rely on the reasonable argument that salient points will concentrate around the door handle and count the number of points at the two possible locations on the left and on right side of the door. The side that contains the highest number of salient points is then used to infer the rotational axis of the door, as illustrated in Fig. 3.

Once the door is detected and its position  $\mathbf{t}$  is obtained as the centroid of the patch, we set its orientation  $\mathbf{o}$  as follows. Let  $\mathbf{n}$  be the normal of the closed door plane and  $\mathbf{z}$  the orientation of the robot, then the orientation of the door is obtained by  $\mathbf{o} = \mathbf{n} \cdot \text{sgn}(\mathbf{n}^T \mathbf{z})$ .

As in the previous case (see section II-C), a new sub-map  $S_j$  is created and added to the vertex set  $V$  of the graph  $\mathcal{G}$ . A new edge  $E_{ij}$  is added to  $E$  that corresponds to the relative position of  $S_j$  with respect to the current sub-map  $S_i$  expressed in  $S_i$  reference frame. The new sub-map is placed at the distance  $d$  behind the closed door in the direction given by  $\mathbf{o}$  and initialized with entity data so as  $S_j = \{P_D; R_D; \emptyset\}$  where  $P_D$  are the door's points that are assumed to be observed from both sides of the door and  $R_D$  the corresponding group. Then it is updated as  $S_j = \{P_D; R_D; U_j\}$  where  $U_j$  is obtained through extrapolation of free space according to the following rule:

$$U_j = \{\mathbf{u} \in \mathbb{R}^2 | f_d(\mathbf{u}) > p_{th}\} \quad (3)$$

where the function  $f_d(\cdot)$  is a bivariate skew-normal (SN) probability density function denoted as  $\mathcal{N}$  and centered at  $\mathbf{t}'$  which is the position of the rotational axis of the door in  $S_j$  reference frame. The family of skew-normal distributions

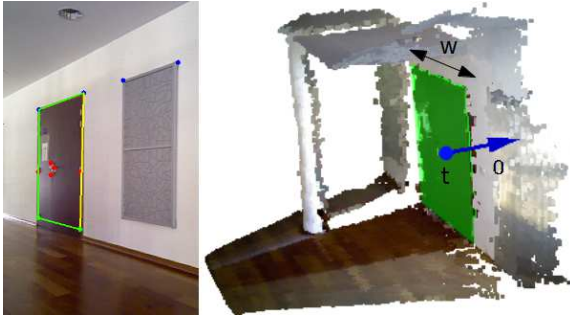


Fig. 3. Door detection example (left) and extracted parameters (right).

[15] constitutes a superset of the normal distribution  $\mathcal{N}$ , being further parameterized by the so-called *shape* parameter that controls the amount of skewness. Following [15], this is achieved by setting  $f_d(\cdot)$  to be a probability density function distributed as  $\mathcal{N}(\mathbf{t}, \Omega, \alpha)$  defined as:

$$f_d(\mathbf{u}) = 2\phi(\mathbf{u})\Phi(\alpha^T \mathbf{u}) \quad (4)$$

where  $\mathbf{u} \in U_j$  is a cell within the local occupancy map,  $\phi$  denotes the normal pdf  $\mathcal{N}(\mathbf{0}, \Omega)$  with covariance matrix  $\Omega = I\sigma^2$ ,  $\Phi$  is the respective cumulative distribution function of  $\phi$  and  $\alpha = (\alpha_1, \alpha_2)^T$  is the parameter vector of the SN pdf that controls the skewness and  $\mathbf{0}$  the origin of  $S_j$  reference frame.

We have chosen to employ the skew-normal pdf due to the fact that doors, in contrast to staircases, cannot be properly modelled with a symmetric form due to their constrained motion as defined by their rotational axis. The SN pdf allows us to capture this asymmetry by regulating the function's skewness parameter vector  $\alpha$  along the orientation of the door and in proportion to the door's width  $w$ . On this basis, skewness is introduced by setting  $\alpha = (0, c \cdot w)^T$ , therefore, the overall parameter vector for the door entity is denoted as  $\mathbf{a}_d = (\mathbf{t}^T, \mathbf{o}^T, \alpha)^T$ . Fig. 3 gives an example of door detection annotated with its distinctive parameters.

### E. Dynamic entity based extrapolation

Through the detection of humans and analysis of their motion history we are able to extrapolate map information with respect to space occupancy based on the fact that a human trajectory always traverses navigable space, which is a reasonable argument for indoor environments. For the detection of the articulated human motion we rely on the use of the OpenNI library [16] and perform map extrapolation strictly within the current sub-map  $S_i$  since the position of a human is dynamic and arbitrary and grounding a new local sub-map is not reasonable. Therefore, we exploit the nature of human navigation in order to infer the presence of areas that are conditionally unoccupied in the current sub-map, in two distinct scenarios. In the first scenario, a human enters or exits the perceptual range of the robot either by entering/exiting a room via a door or by entering/exiting the field of view of the underlying sensor. In both cases, we extrapolate in time (either backward or forward) a short-term human trajectory that in turn allows the extrapolation of free space. In the second scenario, a human upper body is detected while the lower is occluded due to a physical barrier. In that case, extrapolation

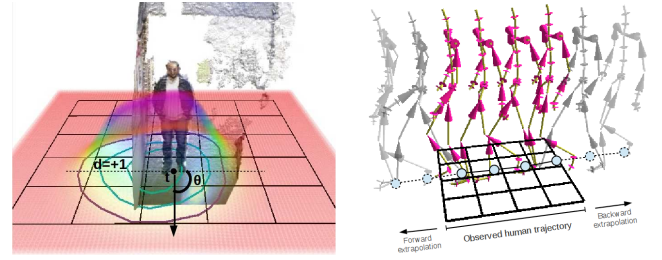


Fig. 4. Left: instance of the atomic space function attributed to a human. Right: trajectory prediction upon entering/exiting the robot's field of view.

of free space is performed by attributing the occluded local area along the human trajectory as unoccupied.

These areas are considered as conditionally unoccupied since the robot would only be able to traverse them in the absence of humans. To address these scenarios, our approach is to treat a human as a non-stationary kernel function which quantifies the notion of personal space introduced by Hall [17] within *proxemics* theory and to utilize this function as a guide for extrapolating the "free space" class, map information along the human trajectory.

Following the description given in section II-A, the first step is to compute the human's pose and class-dependent attributes of the corresponding entity region  $R$ . Towards this end, we collect the dominant human social spacing cues, in particular, (i) the 2D position of the human  $\mathbf{t} = (t_x, t_y)^T \in \mathbb{R}^2$ , (ii) the 2D orientation  $\theta \in [0, 2\pi)$  and (iii) the dominant side  $d \in \{-1, +1\}$  where  $-1, +1$  correspond to the left and right side of the human respectively. The position of the human is taken as the body centroid while the orientation from the torso orientation. The dominant side can be inferred implicitly by the minimum lateral distance of the human from occupied space within the metric map, based on the hypothesis that people maintain a smaller distance from their dominant side and vice versa. In this fashion, the parameter vector for a person is denoted as  $\mathbf{a}_h = (\mathbf{t}^T, \theta, d)^T$ .

In the sequel, we update the current sub-map  $S_i$  as  $S'_i = \{P; R; U'\}$  where  $U' = U \cup U_i$ , and  $U_i$  is obtained through extrapolation of free space via the rule:

$$U_i = \{\mathbf{u} \in \mathbb{R}^2 | f_h(\mathbf{u}) > p_{th}\} \quad (5)$$

where the function  $f_h(\cdot)$  is a bivariate skew-normal (SN) probability density function denoted as  $\mathcal{N}(\mathbf{t}, \Omega, \alpha)$  and  $\mathbf{u} \in U'$  is a cell within the local occupancy map. The  $f_h(\cdot)$  function serves the role of quantifying a human's personal space which we exploit here in order to extrapolate navigable areas along his trajectory. Fig. 4, left, gives a visual example of an individual's personal space as expressed by the function  $f_h(\cdot)$ .

Recalling the earlier mentioned scenarios, forward temporal extrapolation of the human trajectory is applied when the human exits the perceptual field of view of the robot while backward temporal extrapolation is applied upon entry. We perform a linear prediction based on the fact that humans paths are mostly composed of linear segments. Furthermore, since we are only interested in making short-term predictions any changes in human orientation that may occur within the region of interest should have negligible effect. The formula

for obtaining the extrapolated parameter vector for the case of the human entity becomes  $\mathbf{a}_h(t+k) = \mathbf{a}_h(t) + k \cdot \dot{\mathbf{a}}_h(t)$  where  $t$  is the time of the last observed human pose and  $k$  the desired (negative, or positive) prediction time. Fig. 4, right shows the idea of linear trajectory prediction in a synthetic example.

### III. EXPERIMENTS

To evaluate the utility of the proposed approach we have performed a series of indoor mapping experiments using a Neobotix MP-500 robot which employs a two-wheel differential drive mobility system. The robot is equipped with a front-facing 2D laser scanner that has been used in this work as a means for obtaining an accurate estimate of 2D space occupancy. For acquiring RGB-D point clouds of the indoor environment, we have used the Asus XtionPRO Live camera which captures synchronized RGB-D images at a high frame rate. The environment under consideration consists of various spaces with diverse roles, namely, offices, corridors, stairs, reception desk, printing room, meeting rooms, etc.

The experimental scenario is structured as follows. The robot navigates in a supervised mode by a remote operator while encountering various entities (static, quasi-static and dynamic) along its path. To facilitate the subsequent processing of information, during the detection of a human within the scene the robot is halted in order to minimize the interference of scene motion into human motion analysis. The detection of an entity triggers the corresponding class-dependent extrapolation process as described in section II.

*a) Graph extrapolation:* To estimate the capability of our algorithm to add new nodes whenever necessary we have evaluated the performance of our entity detector in the case of static and semi-static entities detection. We use the *recall* ( $R$ ) as a performance measure which is defined as the ratio of true positives to the total number of images where the corresponding entities are present. The overall performance for our experiments is shown in Table I.

TABLE I. Detection performance

Entity class	Recall
Stairs	0.95
Doors	0.67

TABLE II. Extrapolation performance

Entity class	$p_{th}$ threshold		
	0.5	0.6	0.7
Static	0.91	0.93	0.93
Quasi-Static	0.98	0.99	0.99
Dynamic	0.95	0.94	0.93
<b>Overall</b>	<b>0.95</b>	<b>0.95</b>	<b>0.95</b>

In the case of staircases, we attain a high detection performance as their structure is very characteristic within indoor environments, relying on a large 3D form within the scene and therefore being less sensitive to noise. Door detection performance is lower probably due to the requirement of detecting all four corners of the door which can be more challenging within cluttered scenes. Despite this shortcoming, the concerned entities within our environment were consistently detected whenever the robot could have a complete view of the corresponding entity. For both static and quasi-static categories, all instances were detected and the corresponding nodes were added to the global graph so that our approach was able to capture the complete structure of the test environment.

Figure 5 demonstrates the increased capacity of our approach to map an environment through the use of scene context that we exploit to compensate for the lack of information.

It shows graph extrapolation results at the right and ground truth at the left. We see that for all unobserved rooms as for the upper floor there is a corresponding predicted node in the graph. It demonstrates the ability of our algorithm to capture the topology of the environment despite some parts are not accessible at the exploration time. In our case, instead of considering that the environment is only one room, the robot is able to predict the possible existence of three additional rooms and another floor. This definitely increases the environment awareness of the robot and extend the map considerably.

*b) Free space extrapolation:* We have quantitatively evaluated our approach for free space extrapolation using a total of three confidence levels as expressed by the parameter  $p_{th}$ , namely,  $p_{th} \in \{0.5, 0.6, 0.7\}$ . To measure the performance for each confidence level, we use the *precision* score ( $PR$ ) which is defined as the ratio between the number of true positive (TP) predictions of free space and the total number of positive predictions which includes the false positive (FP), i.e.  $PR = TP / (TP + FP)$ . The overall performance as well as for each individual entity class is shown in Table. II.

Clearly, we attain a very high precision in predicting the existence of free space in all different entity classes and across all three confidence levels. These results confirm our hypotheses to a high degree regarding the strong relation that exists between different mapping levels, in this case, between the high-level semantic/social with the lower space occupancy level. Regarding the variation in performance between the chosen confidence thresholds  $p_{th}$ , we could not distinguish a characteristic trend, although one could naturally expect that performance would decrease as  $p_{th}$  decreases. In our experiments, this did not eventually occur because as we increase the area in which we extrapolate free space, new true positives occur along with the false positives and therefore the overall performance remains largely unchanged.

One reason for not using longer-range predictions through the lowering of  $p_{th}$ , is that results become less significant when distance increases. The predicted free space extends beyond

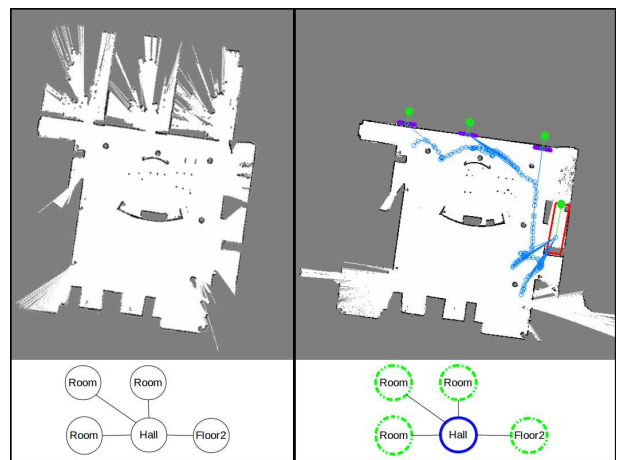


Fig. 5. Left: ground truth map and topology. Top Right: test map of the environment with robot path superimposed. White parts correspond to observed areas, blue nodes to observed sub-maps, green nodes to predicted sub-maps. Red rectangle illustrates staircase while purple lines highlight detected closed doors. Bottom right: Blue circle illustrates observed topology and green dashed circles predicted topology.

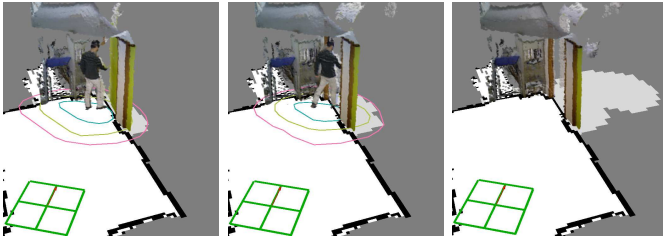


Fig. 6. Local map extrapolation along the observed and the short-term prediction of human trajectory. The robot position is shown as the green cross-mark, extrapolated free map information is shown in light grey, unknown space in dark grey and obstacles in black.

the boundaries of rooms and no longer corresponds to the free space accessible through the entity class that is used for prediction. It highlights one limitation of our approach: we use the high-level semantic/social information to predict free space without accounting for the surrounding environment structure. Nevertheless our method produces globally consistent predictions that definitely increase the quality of the map.

Finally, we provide some qualitative examples of the performance of our approach for different entities. Fig. 6 shows an example of map extrapolation by exploiting the dynamic entity (human) where the robot is placed at the end of corridor when it encounters a human who enters within the field of view of the robot and then exits through a passage. The consecutive frames show the extrapolation result that happens along the observed human trajectory and its short-term prediction after exiting. Figure 7 provides a larger example wherein map extrapolation occurs due to doors detection. The left part shows the corresponding environment when the doors were opened while the right part the mapping output of our approach when the doors were closed. The surface mapped with our approach is up to 10% higher when  $p_{th} = 0.7$  than what would be otherwise obtained and 20% higher when  $p_{th} = 0.5$ . This performance could be even higher within an office environment where there are more semantic cues like doors or humans. This demonstrates the practical utility in performing map extrapolation using our approach, as a way to extend the situation awareness of the robot in realistic conditions.

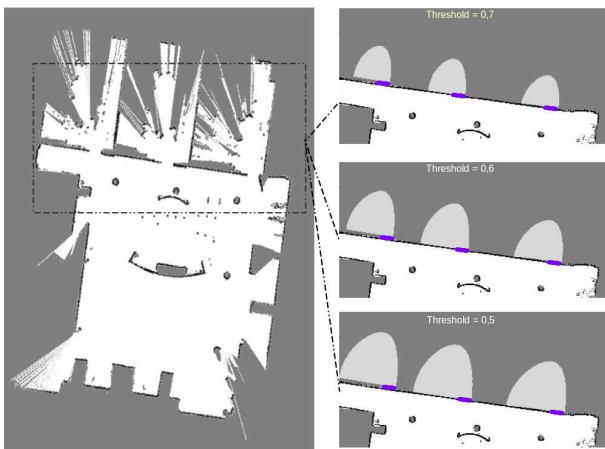


Fig. 7. Map extrapolation using static entities. Ground truth complete map is shown on the left while free space extrapolation is shown on the right in light gray for three confidence thresholds.

## IV. CONCLUSIONS

We have presented a new approach for robot mapping that accounts both for the static as well as the dynamic semantic content of the environment. Our approach exploits the hidden context within the environment at different layers in order to extrapolate map information to unobserved areas, allowing for a significant augmentation in the situation awareness of a robot. We have evaluated the capacity of our approach using a prototype implementation on an indoor robot platform, obtaining very promising results in various scenarios. Among the various directions in which this work could be extended, we believe that through the treatment of additional entity classes we could discover new relations in the various levels of the mapping hierarchy and arrive to a more coherent robot mapping framework, allowing the development of more advanced path-planning behaviours for various applications.

## REFERENCES

- [1] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Int. Conf. on Robotics and Automation*, 2003.
- [2] A. Chapoulie, P. Rives, and D. Filliat, "Topological segmentation of indoors/outdoors sequences of spherical views," in *Int. Conf. on Intelligent Robots and Systems*, 2012.
- [3] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.
- [4] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots—an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359–371, 2007.
- [5] P. Papadakis, A. Spalanzani, and C. Laugier, "Social mapping of human-populated environments by implicit function learning," in *Int. Conf. on Intelligent Robots and Systems*, 2013.
- [6] D. Calisi, L. Iocchi, D. Nardi, C. M. Scalzo, and V. Ziparo, "Contextual navigation and mapping for rescue robots," in *IEEE Int. Workshop on Safety, Security and Rescue Robotics*, 2008.
- [7] S. OCallaghan and F. Ramos, "Gaussian process occupancy maps," *Int. Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, 2012.
- [8] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Int. Conf. on Robotics and Automation*, 2012.
- [9] A. Alempijevic, R. Fitch, and N. Kirchner, "Bootstrapping navigation and path planning using human positional traces," in *Int. Conf. on Robotics and Automation*, 2013.
- [10] S. Grzonka, F. Dijoux, A. Karwath, and W. Burgard, "Mapping indoor environments based on human activity," in *Int. Conf. on Robotics and Automation*, 2010.
- [11] Y. Jiang, H. Koppula, and A. Saxena, "Hallucinated humans as the hidden context for labeling 3d scenes," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2013.
- [12] S. Thrun, "Exploring artificial intelligence in the new millennium," 2003, ch. Robotic Mapping: A Survey, pp. 1–35.
- [13] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.
- [14] J. Shi and C. Tomasi, "Good features to track," in *Int. Conf. on Computer Vision and Pattern Recognition*, 1994.
- [15] A. Azzalini and A. D. Valle, "The multivariate skew-normal distribution," *Biometrika*, vol. 83, no. 4, pp. 715–726, 1996.
- [16] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Int. Conf. on Computer Vision and Pattern Recognition*, 2011.
- [17] E. Hall, *The Hidden Dimension: Man's Use of Space in Public and Private*. The Bodley Head Ltd, 1966.